

[illegible]

[illegible]


```
1 0001 0 MODULE OPC$SECURITY (
2 0002 0 LANGUAGE (BLISS32),
3 0003 0 IDENT = 'V04-000'
4 0004 0 ) =
5 0005 0
6 0006 0 *****
7 0007 0 *
8 0008 0 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 0 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 0 * ALL RIGHTS RESERVED.
11 0011 0 *
12 0012 0 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 0 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 0 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 0 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 0 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 0 * TRANSFERRED.
18 0018 0 *
19 0019 0 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 0 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 0 * CORPORATION.
22 0022 0 *
23 0023 0 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 0 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 0 *
26 0026 0 *
27 0027 0 *****
28 0028 0
29 0029 0 ++
30 0030 0 FACILITY:
31 0031 0
32 0032 0 OPCOM
33 0033 0
34 0034 0 ABSTRACT:
35 0035 0
36 0036 0 This module contains the logic to format security alarm messages.
37 0037 0
38 0038 0 Environment:
39 0039 0
40 0040 0 VAX/VMS operating system.
41 0041 0
42 0042 0 Author:
43 0043 0
44 0044 0 CW Hobbs, from original by Steven T. Jeffreys
45 0045 0
46 0046 0 Creation date:
47 0047 0
48 0048 0 1-Jul-1983 (From OPCOMRQST.B32)
49 0049 0
50 0050 0 MODIFIED BY:
51 0051 0
52 0052 0 V03-007 ACG0449 Andrew C. Goldstein, 30-Aug-1984 16:44
53 0053 0 Tie off SYSUAF fields mod report, since AUTHORIZE isn't
54 0054 0 reporting them. Allow zero length volume labels, and display
55 0055 0 as "(None)".
56 0056 0
57 0057 0 V03-006 MHB0141 Mark Bramhall 2-May-1984
```



```

: 58      0058  0  :
: 59      0059  0  :
: 60      0060  0  :
: 61      0061  0  :
: 62      0062  0  :
: 63      0063  0  :
: 64      0064  0  :
: 65      0065  0  :
: 66      0066  0  :
: 67      0067  0  :
: 68      0068  0  :
: 69      0069  0  :
: 70      0070  0  :
: 71      0071  0  :
: 72      0072  0  :
: 73      0073  0  :
: 74      0074  0  :
: 75      0075  0  :
: 76      0076  0  :
: 77      0077  0  :
: 78      0078  0  :
: 79      0079  0  :
: 80      0080  0  :
: 81      0081  0  :
: 82      0082  0  :
: 83      0083  0  :
: 84      0084  0  :
: 85      0085  0  :
: 86      0086  0  :
: 87      0087  0  :
: 88      0088  0  :

```

```

Allow 0 length node name and password packet strings.
Add SYSUAFF packet to UAF additions.
Correct SYSUAF and NETUAF record processing.
Substitute "(None)" for 0 length node names.
Format password packets using FAO's AF filter.
Correct SYSUAFF packet expansion strings.

V03-005 CWH3005      CW Hobbs      12-Apr-1984
Use the $FAO macro to call fao, rather than direct calls
to SYS$FAO entry.

V03-004 RSH0127      R. Scott Hanna  21-Mar-1984
Modify PUT_PKT_IMGNAM to display (None) when there is no
image name. Modify PUT_PKT_PRIVUSED to display (None) when
no privileges were needed to access the file. Modify FORMAT_VOL
to provide for optional volume set name. Remove PKTTYP EPID
packet from detached loginout messages. Add PKTTYP STATUS to
login failure messages. Remove formatting of record type ACL
messages.

V03-003 RSH0113      R. Scott Hanna  12-Mar-1984
Add the routines to format the security alarm messages.

V03-002 CWH3002      CW Hobbs      16-Sep-1983
Correctly display node name with message.

V03-001 CWH3001      CW Hobbs      30-Jul-1983
Various and sundry things to make OPCOM distributed
across the cluster.

```



```

: 90      0089 1 BEGIN                                ! Start of SECURITY
: 91      0090 1
: 92      0091 1 LIBRARY 'SYSS$LIBRARY:LIB.L32';
: 93      0092 1 LIBRARY 'LIB$:OPCOMLIB';
: 94      0093 1
: 95      0094 1 REQUIRE 'SHRLIB$:NMADEF.R32';
: 96      1394 1
: 97      1395 1 EXTERNAL ROUTINE
: 98      1396 1     CHECK_REQUEST,                   ! Common sanity checks
: 99      1397 1     CLUSMSG_RQCB_SEND,               ! Send an RQCB to the cluster
100      1398 1     DEALLOCATE_RQCB : NOVALUE,         ! Dispose of an RQCB
101      1399 1     FIND_OCD,                         ! Find a given OCD
102      1400 1     FORMAT_MESSAGE,                   ! Format a non-security alarm message
103      1401 1     LOG_MESSAGE,                      ! Write a message to the logfile
104      1402 1     NOTIFY_LISTED_OPERATORS,          ! Notify operators on an operator list
105      1403 1     SEND_REPLY;                       ! Send a reply to the requestor
106      1404 1
107      1405 1 EXTERNAL
108      1406 1     LCL_NODENAME : $BLOCK,             ! Name of local node (DECnet or cluster)
109      1407 1     REQUEST_NUMBER : LONG;             ! Current request #
110      1408 1
111      1409 1 FORWARD ROUTINE
112      1410 1     SECURITY_HANDLER : NOVALUE,         ! Security operator request handler
113      1411 1     FORMAT_ALARM,                      ! Alarm formatting main routine
114      1412 1     VALIDATE_RECORD,                   ! Validate the alarm record
115      1413 1     FORMAT_FIL,                        ! Format file access alarms
116      1414 1     FORMAT_SYSUAF,                     ! Format system UAF alarms
117      1415 1     FORMAT_NETUAF,                     ! Format network UAF alarms
118      1416 1     FORMAT_LOGB,                       ! Format breakin detection alarms
119      1417 1     FORMAT_LOGI,                       ! Format login alarms
120      1418 1     FORMAT_LOGF,                       ! Format login failure alarms
121      1419 1     FORMAT_LOGO,                       ! Format logout alarms
122      1420 1     FORMAT_VOL,                        ! Format volume operation alarms
123      1421 1     PUT_COMMON,                        ! Put common header lines
124      1422 1     PUT_NETUAF_LINE : NOVALUE,          ! Put network UAF line
125      1423 1     PUT_DEVNAM_LINE : NOVALUE,          ! Put device name line
126      1424 1     PUT_SOURCE_LINE,                   ! Put network source line
127      1425 1     PUT_PKT_STATUS,                   ! Put status longword line
128      1426 1     PUT_PKT_PARENTPID,                 ! Put parent PID line
129      1427 1     PUT_PKT_PARENTUN : NOVALUE,         ! Put parent username line
130      1428 1     PUT_PKT_VOLUIC,                   ! Put volume UIC line
131      1429 1     PUT_PKT_IMGNAME : NOVALUE,          ! Put image name line
132      1430 1     PUT_PKT_FILNAM : NOVALUE,           ! Put file name line
133      1431 1     PUT_PKT_FACMOD : NOVALUE,           ! Put file access mode line
134      1432 1     PUT_PKT_PRIVUSED : NOVALUE,         ! Put privileges used line
135      1433 1     PUT_PKT_SYSUAFF : NOVALUE,          ! Put system UAF fields line
136      1434 1     PUT_PKT_PASSWORD,                 ! Put password line
137      1435 1     PUT_PKT_LOGNAM : NOVALUE,           ! Put logical name line
138      1436 1     PUT_PKT_VOLNAM : NOVALUE,           ! Put volume name line
139      1437 1     PUT_PKT_VOLSNAME : NOVALUE,         ! Put volume set name line
140      1438 1     PUT_PKT_VOLPRO : NOVALUE,          ! Put volume protection line
141      1439 1     PUT_PKT_MOUNFLAG : NOVALUE,         ! Put mount flags line
142      1440 1     PUT_PKT_DMOUNFLAG : NOVALUE,       ! Put dismount flags line
143      1441 1     PUT_PKT_STRING : NOVALUE,          ! Put a packet of string data
144      1442 1     PUT_KEYWORDS : NOVALUE,           ! Format and put a packet of keywords
145      1443 1     PUT_STRING : NOVALUE,              ! Put string data
146      1444 1     PUT_STRING_ASCIC : NOVALUE,        ! Put ASCII string data

```



```
147 1445 1 PUT_STRING_ASCID : NOVALUE, : Put ASCII string data
148 1446 1 PUT_NEWLINE : NOVALUE, : Put a CRLF
149 1447 1 STRIP_BLANKS_TRAIL : NOVALUE, : Strip trailing blanks
150 1448 1 STRIP_BLANKS_STRING : NOVALUE; : Strip all blanks
151 1449 1
152 1450 1 BUILTIN
153 1451 1 INSQUE, : Insert entry onto a queue
154 1452 1 REMQUE; : Remove entry from a queue
155 1453 1
156 1454 1 LITERAL
157 1455 1 DATA_LINE_SIZE = 55, : Maximum size of data on a single line
158 1456 1 MAXPKTS_IN_REC = 8; : Maximum number of packets in a record
159 1457 1
160 1458 1 BIND
161 1459 1 COLONS_BUF = UPLIT(BYTE(%ASCIC '::')),
162 1460 1 SPACES_BUF = UPLIT(BYTE(%ASCIC ' ')),
163 1461 1 NONE_BUF = UPLIT(BYTE(%ASCIC '(None)'));
164 1462 1
165 1463 1 STRUCTURE
166 1464 1 ARRAY2 [I,J; M,N,UNIT=4] =
167 1465 1 [M*N*UNIT]
168 1466 1 (ARRAY2+I*N*UNIT+J*UNIT)<0,8*UNIT>;
169 1467 1
170 1468 1 OWN
171 1469 1 :
172 1470 1 : The alarm message buffer and descriptor
173 1471 1 :
174 1472 1 ALARM_BUF : VECTOR[OPCSK_MAXMESSAGE,BYTE],
175 1473 1
176 1474 1 ALARM_DESCR : $BLOCK[DSCSC_S_BLN]
177 1475 1 PRESET([DSCSW_LENGTH] = OPCS_MAXMESSAGE,
178 1476 1 [DSCSB_DTYPE] = DSCSK_DTYPE_T,
179 1477 1 [DSCSB_CLASS] = DSCSK_CLASS_D,
180 1478 1 [DSCSA_POINTER] = ALARM_BUF);
181 1479 1
182 1480 1 :
183 1481 1 : The packet count table contains a count of the number of data packets,
184 1482 1 : (by type), in the security alarm record
185 1483 1 :
186 1484 1 PKT_COUNT_TAB : VECTOR [NSASK_PKTTPNUM,BYTE],
187 1485 1
188 1486 1 :
189 1487 1 : The packet index table contains the first PKT_POINTER_TAB index for
190 1488 1 : each packet type in the security alarm record.
191 1489 1 :
192 1490 1 PKT_INDEX_TAB : VECTOR [NSASK_PKTTPNUM,WORD],
193 1491 1
194 1492 1 :
195 1493 1 : The packet pointer table contains a list of the addresses of all the data
196 1494 1 : packets in the security alarm record. The address of the first packet
197 1495 1 : of a given data type is contained in the n'th entry of the table where
198 1496 1 : n is obtained from the PKT_INDEX_TAB. Each additional packet of the same
199 1497 1 : type is in successive entries.
200 1498 1 :
201 1499 1 PKT_POINTER_TAB : VECTOR [MAXPKTS_IN_REC, LONG],
202 1500 1
203 1501 1 :
```



```
204 1502 1 ! The packet size table contains the minimum and maximum data size of
205 1503 1 ! all data packets.
206 1504 1
207 1505 1 PKT_SIZE_TAB : ARRAY2 [NSASK_PKTTPNUM,2,WORD]
208 1506 1 INITIAL (WORD (
209 1507 1 0,NSASS_PKT_IMGNAM, ! Image name
210 1508 1 4,4, ! File access mode
211 1509 1 4,4, ! Privilege used to access file
212 1510 1 1,NSASS_PKT_FILNAM, ! File name
213 1511 1 1,NSASS_PKT_DEVNAM, ! Device name
214 1512 1 0,NSASS_PKT_LOGNAM, ! Logical name
215 1513 1 0,NSASS_PKT_VOLNAM, ! Volume name
216 1514 1 1,NSASS_PKT_VOLSNAM, ! Volume set name
217 1515 1 0,NSASS_PKT_NODENAM, ! Node name
218 1516 1 1,NSASS_PKT_USERNAM, ! User name
219 1517 1 0,NSASS_PKT_PASSWORD, ! Password
220 1518 1 4,4, ! UIC
221 1519 1 2,2, ! Volume protection
222 1520 1 4,4, ! Mount flags
223 1521 1 2,2, ! Dismount flags
224 1522 1 8,8, ! Node ID
225 1523 1 4,4, ! Extended PID
226 1524 1 8,8, ! System UAF record fields
227 1525 1 4,4)), ! Status longword
228 1526 1
229 1527 1
230 1528 1 ! The following tables describe the packets that may be found in each
231 1529 1 ! type of security auditing record. In addition, it contains the minimum
232 1530 1 ! and maximum number of packets for each type.
233 1531 1
234 1532 1 RECID_FIL_SUCC : VECTOR[13,BYTE]
235 1533 1 INITIAL (BYTE (4,
236 1534 1 NSASK_PKTTP_FACMOD,1,1,
237 1535 1 NSASK_PKTTP_PRIVUSED,1,1,
238 1536 1 NSASK_PKTTP_FILNAM,1,1,
239 1537 1 NSASK_PKTTP_IMGNAM,1,1)),
240 1538 1
241 1539 1 RECID_FIL_FAIL : VECTOR[10,BYTE]
242 1540 1 INITIAL (BYTE (3,
243 1541 1 NSASK_PKTTP_FACMOD,1,1,
244 1542 1 NSASK_PKTTP_FILNAM,1,1,
245 1543 1 NSASK_PKTTP_IMGNAM,1,1)),
246 1544 1
247 1545 1 RECID_SYSUAF_ADD : VECTOR[7,BYTE]
248 1546 1 INITIAL (BYTE (2,
249 1547 1 NSASK_PKTTP_SYSUAFF,1,1,
250 1548 1 NSASK_PKTTP_USERNAM,1,1)),
251 1549 1
252 1550 1 RECID_SYSUAF_DEL : VECTOR[4,BYTE]
253 1551 1 INITIAL (BYTE (1,
254 1552 1 NSASK_PKTTP_USERNAM,1,1)),
255 1553 1
256 1554 1 RECID_SYSUAF_MOD : VECTOR[7,BYTE]
257 1555 1 INITIAL (BYTE (2,
258 1556 1 NSASK_PKTTP_SYSUAFF,1,1,
259 1557 1 NSASK_PKTTP_USERNAM,1,1)),
260 1558 1
```



```
: 261      1559 1  RECID_SYSUAF_COP : VECTOR[7,BYTE]
: 262      1560 1      INITIAL(BYTE (2,
: 263      1561 1      NSASK_PKTTP_SYSUAFF,1,1,
: 264      1562 1      NSASK_PKTTP_USERNAM,2,2)),
: 265      1563 1
: 266      1564 1  RECID_SYSUAF_REN : VECTOR[7,BYTE]
: 267      1565 1      INITIAL(BYTE (2,
: 268      1566 1      NSASK_PKTTP_SYSUAFF,1,1,
: 269      1567 1      NSASK_PKTTP_USERNAM,2,2)),
: 270      1568 1
: 271      1569 1  RECID_NETUAF_ADD : VECTOR[7,BYTE]
: 272      1570 1      INITIAL(BYTE (2,
: 273      1571 1      NSASK_PKTTP_NODENAM,1,1,
: 274      1572 1      NSASK_PKTTP_USERNAM,2,2)),
: 275      1573 1
: 276      1574 1  RECID_NETUAF_DEL : VECTOR[7,BYTE]
: 277      1575 1      INITIAL(BYTE (2,
: 278      1576 1      NSASK_PKTTP_NODENAM,1,1,
: 279      1577 1      NSASK_PKTTP_USERNAM,2,2)),
: 280      1578 1
: 281      1579 1  RECID_NETUAF_MOD : VECTOR[7,BYTE]
: 282      1580 1      INITIAL(BYTE (2,
: 283      1581 1      NSASK_PKTTP_NODENAM,2,2,
: 284      1582 1      NSASK_PKTTP_USERNAM,4,4)),
: 285      1583 1
: 286      1584 1  RECID_LOGB_DIA : VECTOR[7,BYTE]
: 287      1585 1      INITIAL(BYTE (2,
: 288      1586 1      NSASK_PKTTP_PASSWORD,1,1,
: 289      1587 1      NSASK_PKTTP_DEVNAM,1,2)),
: 290      1588 1
: 291      1589 1  RECID_LOGB_LOC : VECTOR[7,BYTE]
: 292      1590 1      INITIAL(BYTE (2,
: 293      1591 1      NSASK_PKTTP_PASSWORD,1,1,
: 294      1592 1      NSASK_PKTTP_DEVNAM,1,2)),
: 295      1593 1
: 296      1594 1  RECID_LOGB_REM : VECTOR[16,BYTE]
: 297      1595 1      INITIAL(BYTE (5,
: 298      1596 1      NSASK_PKTTP_NODEID,1,1,
: 299      1597 1      NSASK_PKTTP_PASSWORD,1,1,
: 300      1598 1      NSASK_PKTTP_DEVNAM,1,1,
: 301      1599 1      NSASK_PKTTP_NODENAM,1,1,
: 302      1600 1      NSASK_PKTTP_USERNAM,1,1)),
: 303      1601 1
: 304      1602 1  RECID_LOGB_NET : VECTOR[13,BYTE]
: 305      1603 1      INITIAL(BYTE (4,
: 306      1604 1      NSASK_PKTTP_NODEID,1,1,
: 307      1605 1      NSASK_PKTTP_PASSWORD,1,1,
: 308      1606 1      NSASK_PKTTP_NODENAM,1,1,
: 309      1607 1      NSASK_PKTTP_USERNAM,1,1)),
: 310      1608 1
: 311      1609 1  RECID_LOGB_DET : VECTOR[7,BYTE]
: 312      1610 1      INITIAL(BYTE (2,
: 313      1611 1      NSASK_PKTTP_PASSWORD,1,1,
: 314      1612 1      NSASK_PKTTP_USERNAM,1,1)),
: 315      1613 1
: 316      1614 1  RECID_LOGI_BAT : VECTOR[1,BYTE]
: 317      1615 1      INITIAL(BYTE (0)),
```



```
: 375      1673 1      NSASK_PKTTP_USERNAM,1,1)),
: 376      1674 1
: 377      1675 1      RECID_LOGF_SUB      : VECTOR[7,BYTE]
: 378      1676 1      INITIAL(BYTE (2,
: 379      1677 1      NSASK_PKTTP_EPID,1,1,
: 380      1678 1      NSASK_PKTTP_STATUS,1,1)),
: 381      1679 1
: 382      1680 1      RECID_LOGF_DET      : VECTOR[7,BYTE]
: 383      1681 1      INITIAL(BYTE (2,
: 384      1682 1      NSASK_PKTTP_STATUS,1,1,
: 385      1683 1      NSASK_PKTTP_USERNAM,1,1)),
: 386      1684 1
: 387      1685 1      RECID_LOGO_BAT      : VECTOR[1,BYTE]
: 388      1686 1      INITIAL(BYTE (0)),
: 389      1687 1
: 390      1688 1      RECID_LOGO_DIA      : VECTOR[4,BYTE]
: 391      1689 1      INITIAL(BYTE (1,
: 392      1690 1      NSASK_PKTTP_DEVNAM,1,2)),
: 393      1691 1
: 394      1692 1      RECID_LOGO_LOC      : VECTOR[4,BYTE]
: 395      1693 1      INITIAL(BYTE (1,
: 396      1694 1      NSASK_PKTTP_DEVNAM,1,2)),
: 397      1695 1
: 398      1696 1      RECID_LOGO_REM      : VECTOR[13,BYTE]
: 399      1697 1      INITIAL(BYTE (4,
: 400      1698 1      NSASK_PKTTP_NODEID,1,1,
: 401      1699 1      NSASK_PKTTP_DEVNAM,1,1,
: 402      1700 1      NSASK_PKTTP_NODENAM,1,1,
: 403      1701 1      NSASK_PKTTP_USERNAM,1,1)),
: 404      1702 1
: 405      1703 1      RECID_LOGO_NET      : VECTOR[10,BYTE]
: 406      1704 1      INITIAL(BYTE (3,
: 407      1705 1      NSASK_PKTTP_NODEID,1,1,
: 408      1706 1      NSASK_PKTTP_NODENAM,1,1,
: 409      1707 1      NSASK_PKTTP_USERNAM,1,1)),
: 410      1708 1
: 411      1709 1      RECID_LOGO_SUB      : VECTOR[4,BYTE]
: 412      1710 1      INITIAL(BYTE (1,
: 413      1711 1      NSASK_PKTTP_EPID,1,1)),
: 414      1712 1
: 415      1713 1      RECID_LOGO_DET      : VECTOR[1,BYTE]
: 416      1714 1      INITIAL(BYTE (0)),
: 417      1715 1
: 418      1716 1      RECID_VOL_MOU      : VECTOR[25,BYTE]
: 419      1717 1      INITIAL(BYTE (8,
: 420      1718 1      NSASK_PKTTP_UIC,1,1,
: 421      1719 1      NSASK_PKTTP_VOLPRO,1,1,
: 422      1720 1      NSASK_PKTTP_MOUFLG,1,1,
: 423      1721 1      NSASK_PKTTP_IMGNAM,1,1,
: 424      1722 1      NSASK_PKTTP_DEVNAM,1,1,
: 425      1723 1      NSASK_PKTTP_LOGNAM,1,1,
: 426      1724 1      NSASK_PKTTP_VOLNAM,1,1,
: 427      1725 1      NSASK_PKTTP_VOLSNAM,0,1)),
: 428      1726 1
: 429      1727 1      RECID_VOL_DMOU      : VECTOR[19,BYTE]
: 430      1728 1      INITIAL(BYTE (6,
: 431      1729 1      NSASK_PKTTP_DMOUFLG,1,1,
```

```

: 432      1730 1      NSASK_PKTTYP_IMG NAM,1,1,
: 433      1731 1      NSASK_PKTTYP_DEV NAM,1,1,
: 434      1732 1      NSASK_PKTTYP_LOG NAM,1,1,
: 435      1733 1      NSASK_PKTTYP_VOL NAM,1,1,
: 436      1734 1      NSASK_PKTTYP_VOLS NAM,0,1)),
: 437      1735 1
: 438      1736 1
: 439      1737 1
: 440      1738 1
: 441      1739 1
: 442      1740 1      ! The following tables contain the number of subtypes of each record
: 443      1741 1      ! type and a pointer to the table which describes the record content.
: 444      1742 1      !
: 445      1743 1      !
: 446      1744 1      !
: 447      1745 1      !
: 448      1746 1      !
: 449      1747 1      !
: 450      1748 1      !
: 451      1749 1      !
: 452      1750 1      !
: 453      1751 1      !
: 454      1752 1      !
: 455      1753 1      !
: 456      1754 1      !
: 457      1755 1      !
: 458      1756 1      !
: 459      1757 1      !
: 460      1758 1      !
: 461      1759 1      !
: 462      1760 1      !
: 463      1761 1      !
: 464      1762 1      !
: 465      1763 1      !
: 466      1764 1      !
: 467      1765 1      !
: 468      1766 1      !
: 469      1767 1      !
: 470      1768 1      !
: 471      1769 1      !
: 472      1770 1      !
: 473      1771 1      !
: 474      1772 1      !
: 475      1773 1      !
: 476      1774 1      !
: 477      1775 1      !
: 478      1776 1      !
: 479      1777 1      !
: 480      1778 1      !
: 481      1779 1      !
: 482      1780 1      !
: 483      1781 1      !
: 484      1782 1      !
: 485      1783 1      !
: 486      1784 1      !
: 487      1785 1      !
: 488      1786 1      !

RECTYP_FIL      : VECTOR [NSASK_RECTYPNUM_FIL+1]
                  INITIAL (NSASK_RECTYPNUM_FIL,
                           RECID_FIL_SUCC,
                           RECID_FIL_FAIL),

RECTYP_SYSUAF   : VECTOR [NSASK_RECTYPNUM_SYSUAF+1]
                  INITIAL (NSASK_RECTYPNUM_SYSUAF,
                           RECID_SYSUAF_ADD,
                           RECID_SYSUAF_DEL,
                           RECID_SYSUAF_MOD,
                           RECID_SYSUAF_COP,
                           RECID_SYSUAF_REN),

RECTYP_NETUAF   : VECTOR [NSASK_RECTYPNUM_NETUAF+1]
                  INITIAL (NSASK_RECTYPNUM_NETUAF,
                           RECID_NETUAF_ADD,
                           RECID_NETUAF_DEL,
                           RECID_NETUAF_MOD),

RECTYP_LOGB     : VECTOR [NSASK_RECTYPNUM_LOGB+1]
                  INITIAL (NSASK_RECTYPNUM_LOGB,
                           RECID_LOGB_DIA,
                           RECID_LOGB_LOC,
                           RECID_LOGB_REM,
                           RECID_LOGB_NET,
                           RECID_LOGB_DET),

RECTYP_LOGI     : VECTOR [NSASK_RECTYPNUM_LOGI+1]
                  INITIAL (NSASK_RECTYPNUM_LOGI,
                           RECID_LOGI_BAT,
                           RECID_LOGI_DIA,
                           RECID_LOGI_LOC,
                           RECID_LOGI_REM,
                           RECID_LOGI_NET,
                           RECID_LOGI_SUB,
                           RECID_LOGI_DET),

RECTYP_LOGF     : VECTOR [NSASK_RECTYPNUM_LOGF+1]
                  INITIAL (NSASK_RECTYPNUM_LOGF,
                           RECID_LOGF_BAT,
                           RECID_LOGF_DIA,
                           RECID_LOGF_LOC,
                           RECID_LOGF_REM,
                           RECID_LOGF_NET,
                           RECID_LOGF_SUB,
                           RECID_LOGF_DET),

```



```

: 489      1787 1  RECTYP_LOGO      : VECTOR [NSASK_RECTYPNUM_LOGO+1]
: 490      1788 1  INITIAL (NSASK_RECTYPNUM_LOGO,
: 491      1789 1  RECID_LOGO_BAT,
: 492      1790 1  RECID_LOGO_DIA,
: 493      1791 1  RECID_LOGO_LOC,
: 494      1792 1  RECID_LOGO_REM,
: 495      1793 1  RECID_LOGO_NET,
: 496      1794 1  RECID_LOGO_SUB,
: 497      1795 1  RECID_LOGO_DET),
: 498
: 499      1797 1  RECTYP_VOL      : VECTOR [NSASK_RECTYPNUM_VOL+1]
: 500      1798 1  INITIAL (NSASK_RECTYPNUM_VOL,
: 501      1799 1  RECID_VOL_MOU,
: 502      1800 1  RECID_VOL_DMOU),
: 503      1801 1
: 504      1802 1
: 505      1803 1  | The following table contains the number of record types and a pointer
: 506      1804 1  | to the record subtype table.
: 507      1805 1  |
: 508      1806 1  RECTYP      : VECTOR [NSASK_RECTYPNUM+1]
: 509      1807 1  INITIAL (NSASK_RECTYPNUM,
: 510      1808 1  RECTYP_FIL,
: 511      1809 1  RECTYP_SYSUAF,
: 512      1810 1  RECTYP_NETUAF,
: 513      1811 1  RECTYP_LOGB,
: 514      1812 1  RECTYP_LOGI,
: 515      1813 1  RECTYP_LOGF,
: 516      1814 1  RECTYP_LOGO,
: 517      1815 1  RECTYP_VOL);

```



```

: 519 1816 1 GLOBAL ROUTINE SECURITY_HANDLER (BUFFER_DESC) : NOVALUE =
: 520 1817 1
: 521 1818 1 ++
: 522 1819 1 Functional description:
: 523 1820 1
: 524 1821 1 This routine is the handler for all SECURITY messages received by OPCOM.
: 525 1822 1
: 526 1823 1
: 527 1824 1 Input:
: 528 1825 1
: 529 1826 1 BUFFER_DESC : The address of a quadword buffer descriptor that
: 530 1827 1 describes the buffer containing the message.
: 531 1828 1
: 532 1829 1 Implicit Input:
: 533 1830 1
: 534 1831 1 None.
: 535 1832 1
: 536 1833 1 Output:
: 537 1834 1
: 538 1835 1 None.
: 539 1836 1
: 540 1837 1 Implicit output:
: 541 1838 1
: 542 1839 1 Some accounting data will be updated
: 543 1840 1 to reflect the receipt of the message.
: 544 1841 1
: 545 1842 1 Side effects:
: 546 1843 1
: 547 1844 1 None.
: 548 1845 1
: 549 1846 1 Routine value:
: 550 1847 1
: 551 1848 1 None.
: 552 1849 1 --
: 553 1850 1
: 554 1851 2 BEGIN ! Start of REQUEST_HANDLER
: 555 1852 2
: 556 1853 2 MAP
: 557 1854 2 BUFFER_DESC : $REF_BBLOCK;
: 558 1855 2
: 559 1856 2 LOCAL
: 560 1857 2 MESSAGE_VECTOR : VECTOR [7, LONG], ! Message info
: 561 1858 2 RQCB : $REF_BBLOCK, ! RQCB data structure
: 562 1859 2 OCD : $REF_BBLOCK, ! OCD data structure
: 563 1860 2 MSG : $REF_BBLOCK, ! Pointer to user request
: 564 1861 2 ON_BUF : VECTOR [64, BYTE], ! Buffer for preposition (" on " node)
: 565 1862 2 ON_DSC : VECTOR [2, LONG] ! Desc for preposition
: 566 1863 2 INITIAL (64, ON_BUF),
: 567 1864 2 FOUND : LONG, ! Boolean
: 568 1865 2 SCOPE : LONG, ! Scope of request
: 569 1866 2 SCOPE_LIMIT : LONG, ! Loop control
: 570 1867 2 STATUS : LONG;
: 571 1868 2
: 572 1869 2
: 573 1870 2
: 574 1871 2 Make sure there is enough data in the request.
: 575 1872 2
```



```

576 1873 3 IF .BUFFER_DESC[DSC$W_LENGTH] LSS (OPC$K_COMHDRSIZ + OPC$K_SECURITY_MIN_SIZE)
577 1874 2 THEN
578 1875 2 RETURN;
579 1876 2 ! Ignore the request
580 1877 2 ! Do some common sanity checks.
581 1878 2
582 1879 2 IF NOT CHECK_REQUEST (.BUFFER_DESC, RQCB)
583 1880 2 THEN
584 1881 2 RETURN;
585 1882 2 MESSAGE_VECTOR[0] = 0;
586 1883 2 ! Assume no errors
587 1884 2 ! See if the requestor is issuing this request on another's behalf.
588 1885 2 ! If so, and the requestor does not have the privilege to do so,
589 1886 2 ! then dismiss the request.
590 1887 2
591 1888 2 IF .RQCB[RQCB_L_SENDERUIC] NEQ .RQCB[RQCB_L_UIC]
592 1889 2 THEN
593 1890 3 IF (NOT .SBBLOCK[RQCB [RQCB_L_PRIVMASK1], PRV$V_OPER])
594 1891 3 THEN
595 1892 3 IF NOT ((.SBBLOCK[RQCB[RQCB_L_SENDERUIC], 2,0,16,0] EQL .SBBLOCK[RQCB[RQCB_L_UIC], 2,0,16,0]) AND
596 1893 3 (.SBBLOCK[RQCB[RQCB_L_PRIVMASK1], PRV$V_GROUP]))
597 1894 3 THEN
598 1895 3 BEGIN
599 1896 3 MESSAGE_VECTOR[0] = OPC$_ILLRQST;
600 1897 3 MESSAGE_VECTOR[1] = 0;
601 1898 3 END;
602 1899 2
603 1900 2 ! Create a descriptor within the RQCB to point to the request text.
604 1901 2
605 1902 2 MSG = .BUFFER_DESC[DSC$A_POINTER] + OPC$K_COMHDRSIZ;
606 1903 2 RQCB[RQCB_L_TEXT_LEN] = .MSG[OPC$W_SECURITY_LENGTH];
607 1904 3 IF (.RQCB[RQCB_L_TEXT_LEN] GTR 0)
608 1905 3 THEN
609 1906 3 BEGIN
610 1907 3 ! Create a buffer for the request text and copy the text to the buffer.
611 1908 3
612 1909 3 IF NOT (STATUS = OPC$GET_VM(RQCB[RQCB_L_TEXT_LEN], RQCB[RQCB_L_TEXT_PTR]))
613 1910 4 THEN
614 1911 4 BEGIN
615 1912 4 DEALLOCATE_RQCB(.RQCB);
616 1913 4 RETURN;
617 1914 4 END;
618 1915 3 CH$MOVE(.RQCB[RQCB_L_TEXT_LEN], MSG[OPC$T_SECURITY_TEXT], .RQCB[RQCB_L_TEXT_PTR]);
619 1916 3 END
620 1917 3 ELSE
621 1918 2 BEGIN
622 1919 2 ! There is no request text. Inform the requestor that this is not allowed.
623 1920 2
624 1921 2 MESSAGE_VECTOR[0] = OPC$_ILLRQST;
625 1922 2 MESSAGE_VECTOR[1] = 0;
626 1923 2 END;
627 1924 2
628 1925 2 ! Find an OCD that can handle this request. The OCD is selected
629 1926 2 ! according to the SCOPE and UIC of the requestor. If the SCOPE
630 1927 2 ! is unspecified, then look for operator coverage starting in the
631 1928 2
632 1929 2

```



```

633 1930 2 ! least privileged scope and continuing to the most privileged.
634 1931 2 ! If no OCD is found, then dismiss the request.
635 1932 2
636 1933 3 IF (.RQCB[RQCB_B_SCOPE] EQL OPC$K_UNSPEC)
637 1934 2 THEN
638 1935 2 SCOPE_LIMIT = OPC$K_SYSTEM
639 1936 2 ELSE
640 1937 2 SCOPE_LIMIT = .RQCB[RQCB_B_SCOPE];
641 1938 2 FOUND = FALSE;
642 1939 2 SCOPE = .RQCB[RQCB_B_SCOPE];
643 1940 2 WHILE (.SCOPE GEQ SCOPE_LIMIT) AND (NOT .FOUND) DO
644 1941 3 IF NOT (FOUND = FIND_OCD(.SCOPE, .RQCB[RQCB_L_UIC], OCD))
645 1942 2 THEN
646 1943 2 SCOPE = .SCOPE - 1;
647 1944 2 IF NOT .FOUND
648 1945 2 THEN
649 1946 3 BEGIN
650 1947 3 MESSAGE_VECTOR[0] = OPC$_NOPERATOR; ! No operator coverage
651 1948 3 MESSAGE_VECTOR[1] = 0;
652 1949 3 END;
653 1950 2
654 1951 2 ! If there is an error message to output,
655 1952 2 ! do so and dismiss the request.
656 1953 2
657 1954 2 IF .MESSAGE_VECTOR[0] NEQ 0
658 1955 2 THEN
659 1956 3 BEGIN
660 1957 3 FORMAT_MESSAGE(.RQCB, MESSAGE_VECTOR);
661 1958 3 SEND_REPLY(.RQCB, MESSAGE_VECTOR);
662 1959 3 DEALLOCATE_RQCB(.RQCB);
663 1960 3 RETURN;
664 1961 2 END;
665 1962 2
666 1963 2 ! Set the scope of the request.
667 1964 2
668 1965 2 RQCB[RQCB_L_OCD] = .OCD; ! Save OCD address
669 1966 2 RQCB[RQCB_B_SCOPE] = .OCD[OCD_B_SCOPE]; ! Set request scope
670 1967 2
671 1968 2 ! Get the preposition for the node name
672 1969 2
673 1970 2 IF .LCL_NODENAME[DSC$W_LENGTH] NEQ 0
674 1971 2 THEN
675 1972 3 BEGIN
676 1973 4 IF NOT (STATUS = $GETMSG(MSGID=OPC$_ON_NODE, MSGLEN=ON_DSC, BUFADR=ON_DSC, FLAGS=1))
677 1974 3 THEN
678 1975 3 $signal_stop (.STATUS);
679 1976 3 END
680 1977 2 ELSE
681 1978 2 ON_DSC[0] = 0;
682 1979 2
683 1980 2 ! Format the alarm message
684 1981 2
685 1982 2 STATUS = FORMAT_ALARM(.RQCB[RQCB_L_TEXT_PTR], .RQCB[RQCB_L_TEXT_LEN], MESSAGE_VECTOR[5]);
686 1983 2
687 1984 2 ! Format the request message. Note that messages from users who do not
688 1985 2 ! have security privilege are marked as counterfeit.
689 1986 2
```



```

: 690      1987 2 MESSAGE_VECTOR[1] = 0;                                ! Use current time in stamp
: 691      1988 IF .SBB[LOCK[RQCB[RQCB_L_PRIVMASK1], PRV$V_SECURITY] AND .STATUS
: 692      1989 THEN
: 693      1990 BEGIN
: 694      1991 MESSAGE_VECTOR[0] = OPC$ SECUR_ALARM;
: 695      1992 MESSAGE_VECTOR[2] = ON_DSC;                                ! The " on " field
: 696      1993 MESSAGE_VECTOR[3] = .LCL_NODENAME[DSC$W_LENGTH];          ! Length of nodename
: 697      1994 MESSAGE_VECTOR[4] = .LCL_NODENAME[DSC$A_POINTER];        ! Address of nodename
: 698      1995 END
: 699      1996 ELSE
: 700      1997 BEGIN
: 701      1998 MESSAGE_VECTOR[0] = OPC$ COUNTERFEIT;
: 702      1999 MESSAGE_VECTOR[2] = .RQCB[RQCB_W_USERNAMELEN];          ! Length of username
: 703      2000 MESSAGE_VECTOR[3] = RQCB[RQCB_T_USERNAME];              ! Address of username
: 704      2001 MESSAGE_VECTOR[4] = ON_DSC;                                ! The " on " field
: 705      2002 MESSAGE_VECTOR[5] = .LCL_NODENAME[DSC$W_LENGTH];          ! Length of nodename
: 706      2003 MESSAGE_VECTOR[6] = .LCL_NODENAME[DSC$A_POINTER];        ! Address of nodename
: 707      2004 END;
: 708      2005
: 709      2006 FORMAT_MESSAGE(.RQCB, MESSAGE_VECTOR);                    ! Format the security alarm message
: 710      2007
: 711      2008 ! Tell the cluster, notify operators and log the request.
: 712      2009
: 713      2010 CLUSMSG RQCB SEND(-1, CLM_SECURITY, .RQCB);
: 714      2011 NOTIFY LISTED_OPERATORS(.RQCB);
: 715      2012 LOG_MESSAGE(.RQCB);
: 716      2013
: 717      2014 DEALLOCATE_RQCB(.RQCB);
: 718      2015
: 719      2016 1 END;                                                    ! End of SECURITY_HANDLER
```

```

                                .TITLE  OPC$SECURITY
                                .IDENT   \V04-000\
                                .PSECT   $SPLIT$,NOWRT,NOEXE,2
                                3A 3A 02 00000 P.AAA: .ASCII <2>\::\
                                00003 .BLKB 1
                                20 20 02 00004 P.AAB: .ASCII <2>\ \
                                00007 .BLKB 1
29 65 6E 6F 4E 28 06 00008 P.AAC: .ASCII <6>\(None)\
                                .PSECT   $OWNS$,NOEXE,2
                                00000 ALARM_BUF:
                                .BLKB 2048
                                0800 00800 ALARM_DESCR:
                                .WORD 2048
                                02 0E 00802 .BYTE 14, 2
00000000' 00804 .ADDRESS ALARM_BUF
                                00808 PKT_COUNT TAB:
                                .BLKB 19
                                0081B .BLKB 1
                                0081C PKT_INDEX TAB:
                                .BLKB 38
                                00842 .BLKB 2
```



```
00844 PKT_POINTER TAB:
00864 PKT_SIZE TAB:
00878
0088C
008A0
008B0 RECID_FIL_SUCC:
008BD
008C0 RECID_FIL_FAIL:
008CA
008CC RECID_SYSAF_ADD:
008D3
008D4 RECID_SYSAF_DEL:
008D8 RECID_SYSAF_MOD:
008DF
008E0 RECID_SYSAF_COP:
008E7
008E8 RECID_SYSAF_REN:
008EF
008F0 RECID_NETUAF_ADD:
008F7
008F8 RECID_NETUAF_DEL:
008FF
00900 RECID_NETUAF_MOD:
00907
00908 RECID_LOGB_DIA:
0090F
00910 RECID_LOGB_LOC:
00917
00918 RECID_LOGB_REM:
00927
00928 RECID_LOGB_NET:
00935
00938 RECID_LOGB_DET:
0093F
00940 RECID_LOGI_BAT:
00941
00944 RECID_LOGI_DIA:
00948 RECID_LOGI_LOC:

000F 0001 01BC 0001 0004 0004 0004 0004 01BC 0000
000C 0001 0006 0000 000C 0001 000C 0000 00FF 0000
0002 0002 0004 0004 0002 0002 0004 0004 001F 0000
      01 01 01 01 01 04 01 01 03 01 01 02 04
      01 01 01 01 01 04 01 01 02 03
      01 01 0A 01 01 12 02
      01 01 0A 01 01 12 02
      02 02 0A 01 01 12 02
      02 02 0A 01 01 12 02
      02 02 0A 01 01 09 02
      02 02 0A 01 01 09 02
      04 04 0A 02 02 09 02
      02 01 05 01 01 0B 02
      02 01 05 01 01 0B 02
01 0A 01 01 09 01 01 05 01 01 0B 01 01 10 05
      01 01 0A 01 01 09 01 01 0B 01 01 10 04
      01 01 0A 01 01 0B 02
      00
      02 01 05 01
      02 01 05 01
```


01	01	0A	01	01	09	01	01	05	01	01	10	04	0094C	RECID_LOGI	REM:	.BYTE	1, 5, 1, 2	:					
													00959			.BYTE	4, 16, 1, 1, 5, 1, 1, 9, 1, 1, 10, 1, 1	:					
		01	01	0A	01	01	09	01	01	10	03	0095C	RECID_LOGI	NET:		.BLKB	3	:					
												00966				.BYTE	3, 16, 1, 1, 9, 1, 1, 10, 1, 1	:					
									01	01	11	01	00968	RECID_LOGI	SUB:		.BLKB	2	:				
									01	01	0A	01	0096C	RECID_LOGI	DET:		.BYTE	1, 17, 1, 1	:				
									01	01	13	01	00970	RECID_LOGF	BAT:		.BYTE	1, 10, 1, 1	:				
																.BYTE	1, 19, 1, 1	:					
						02	01	05	01	01	13	02	00974	RECID_LOGF	DIA:		.BYTE	1, 19, 1, 1	:				
													0097B				.BYTE	2, 19, 1, 1, 5, 1, 2	:				
													0097C	RECID_LOGF	LOC:		.BLKB	1	:				
																.BYTE	2, 19, 1, 1, 5, 1, 2	:					
01	0A	01	01	09	01	01	05	01	01	13	01	01	00983				.BLKB	1	:				
													00984	RECID_LOGF	REM:		.BYTE	5, 16, 1, 1, 19, 1, 1, 5, 1, 1, 9, 1, 1, -	:				
													01	00993				10, 1, 1	:				
		01	01	0A	01	01	09	01	01	13	01	01	10	04	00994	RECID_LOGF	NET:		.BYTE	4, 16, 1, 1, 19, 1, 1, 9, 1, 1, 10, 1, 1	:		
													009A1				.BLKB	3	:				
									01	01	13	01	01	11	02	009A4	RECID_LOGF	SUB:		.BYTE	2, 17, 1, 1, 19, 1, 1	:	
													009AB				.BLKB	1	:				
									01	01	0A	01	01	13	02	009AC	RECID_LOGF	DET:		.BYTE	2, 19, 1, 1, 10, 1, 1	:	
													009B3				.BLKB	1	:				
												00	009B4	RECID_LOGO	BAT:		.BYTE	0	:				
																.BLKB	3	:					
													009B5				.BLKB	3	:				
									02	01	05	01	009B8	RECID_LOGO	DIA:		.BYTE	1, 5, 1, 2	:				
																.BYTE	1, 5, 1, 2	:	:				
									02	01	05	01	009BC	RECID_LOGO	LOC:		.BYTE	1, 5, 1, 2	:				
01	01	0A	01	01	09	01	01	05	01	01	10	04	009C0	RECID_LOGO	REM:		.BYTE	4, 16, 1, 1, 5, 1, 1, 9, 1, 1, 10, 1, 1	:				
													009CD				.BLKB	3	:				
													009D0	RECID_LOGO	NET:		.BYTE	3, 16, 1, 1, 9, 1, 1, 10, 1, 1	:				
													009DA				.BLKB	2	:				
													009DC	RECID_LOGO	SUB:		.BYTE	1, 17, 1, 1	:				
													00	009E0	RECID_LOGO	DET:		.BYTE	0	:			
																.BLKB	3	:					
01	05	01	01	01	01	01	0E	01	01	0D	01	01	0C	08	009E1								
															009E4	RECID_VOL	MOU:		.BYTE	8, 12, 1, 1, 13, 1, 1, 14, 1, 1, 1, 1, 1, -	:		
															009F3				.BLKB	5, 1, 1, 6, 1, 1, 7, 1, 1, 8, 0, 1	:		
															009FD				.BLKB	3	:		
01	07	01	01	06	01	01	05	01	01	01	01	01	0F	06	00A00	RECID_VOL	DMOU:		.BYTE	6, 15, 1, 1, 1, 1, 1, 5, 1, 1, 6, 1, 1, -	:		
															00A0F				.BLKB	7, 1, 1, 8, 0, 1	:		
															00A13				.BLKB	1	:		


```
00000002 00A14 RECTYP_FIL:
                                .LONG 2
00000000' 00000000' 00A18 .ADDRESS RECID_FIL_SUCC, RECID_FIL_FAIL
00000005 00A20 RECTYP_SYSUAF:
                                .LONG 5
00000000' 00000000' 00000000' 00000000' 00000000' 00A24 .ADDRESS RECID_SYSUAF_ADD, RECID_SYSUAF_DEL, -
                                                                RECID_SYSUAF_MOD, RECID_SYSUAF_COP, -
                                                                RECID_SYSUAF_REN
00000003 00A38 RECTYP_NETUAF:
                                .LONG 3
00000000' 00000000' 00000000' 00A3C .ADDRESS RECID_NETUAF_ADD, RECID_NETUAF_DEL, -
                                                                RECID_NETUAF_MOD
00000005 00A48 RECTYP_LOGB:
                                .LONG 5
00000000' 00000000' 00000000' 00000000' 00000000' 00A4C .ADDRESS RECID_LOGB_DIA, RECID_LOGB_LOC, -
                                                                RECID_LOGB_REM, RECID_LOGB_NET, -
                                                                RECID_LOGB_DET
00000007 00A60 RECTYP_LOGI:
                                .LONG 7
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00A64 .ADDRESS RECID_LOGI_BAT, RECID_LOGI_DIA, -
00000000' 00A7C .ADDRESS RECID_LOGI_LOC, RECID_LOGI_REM, -
                                                                RECID_LOGI_NET, RECID_LOGI_SUB, -
                                                                RECID_LOGI_DET
00000007 00A80 RECTYP_LOGF:
                                .LONG 7
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00A84 .ADDRESS RECID_LOGF_BAT, RECID_LOGF_DIA, -
00000000' 00A9C .ADDRESS RECID_LOGF_LOC, RECID_LOGF_REM, -
                                                                RECID_LOGF_NET, RECID_LOGF_SUB, -
                                                                RECID_LOGF_DET
00000007 00AA0 RECTYP_LOGO:
                                .LONG 7
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00AA4 .ADDRESS RECID_LOGO_BAT, RECID_LOGO_DIA, -
00000000' 00ABC .ADDRESS RECID_LOGO_LOC, RECID_LOGO_REM, -
                                                                RECID_LOGO_NET, RECID_LOGO_SUB, -
                                                                RECID_LOGO_DET
00000002 00AC0 RECTYP_VOL:
                                .LONG 2
00000000' 00000000' 00AC4 .ADDRESS RECID_VOL_MOU, RECID_VOL_DMOU
00000000' 00000008' 00ACC RECTYP: .LONG 8
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00AD0 .ADDRESS RECTYP_FIL, RECTYP_SYSUAF, -
00000000' 00000000' 00AE8 .ADDRESS RECTYP_NETUAF, RECTYP_LOGB, RECTYP_LOGI, -
                                                                RECTYP_LOGF, RECTYP_LOGO, RECTYP_VOL
```

COLONS_BUF=
SPACES_BUF=
NONE_BUF=

P.AAA
P.AAB
P.AAC

.EXTRN CHECK_REQUEST, CLUSMSG_RQCB_SEND
.EXTRN DEALLOCATE_RQCB
.EXTRN FIND_OCD, FORMAT_MESSAGE
.EXTRN LOG_MESSAGE, NOTIFY_LISTED_OPERATORS
.EXTRN SEND_REPLY, LCL_NODENAME
.EXTRN REQUEST_NUMBER, OPC\$GET_VM
.EXTRN SYS\$GETMSG, LIB\$STOP

.PSECT \$CODE\$,NOWRT,2

07FC 00000

.ENTRY SECURITY_HANDLER, Save R2,R3,R4,R5,R6,R7,- ; 1816

		5A	0000G	CF	9E	00002	MOVAB	R8,R9,R10	
		5E	94	AE	9E	00007	MOVAB	LCL NODENAME, R10	
08		AE	40	8F	9A	0000B	MOVZBL	-108(SP), SP	1851
0C		AE	10	AE	9E	00010	MOVAB	#64, ON_DSC	
		52	04	AC	D0	00015	MOVL	ON BUF, -ON_DSC+4	1873
0042		8F		62	B1	00019	CMPW	BUFFER_DEST, R2	
				01	1E	0001E	BGEQU	(R2), #66	
					04	00020	RET	1\$	
			4004	8F	BB	00021	PUSHR	#^M<R2, SP>	1879
0000G		CF		02	FB	00025	CALLS	#2, CHECK_REQUEST	
		01		50	E8	0002A	BLBS	R0, 2\$	
					04	0002D	RET		
			50	AE	D4	0002E	CLRL	MESSAGE_VECTOR	1882
		56		6E	D0	00031	MOVL	R0CB, R8	1888
		50	38	A6	9E	00034	MOVAB	56(R6), R0	
		57	68	A6	9E	00038	MOVAB	104(R6), R7	
		67		60	D1	0003C	CMPL	(R0), (R7)	
				1B	13	0003F	BEQL	4\$	
16		32		02	E0	00041	BBS	#2, 50(R6), 4\$	1890
		02		A0	B1	00046	CMPW	2(R0), 2(R7)	1892
				04	12	0004B	BNEQ	3\$	
			31	A6	E8	0004D	BLBS	49(R6), 4\$	1893
		50		8F	D0	00051	MOVL	#360572, MESSAGE_VECTOR	1896
			54	AE	D4	00059	CLRL	MESSAGE_VECTOR+4	1897
52		04		26	C1	0005C	ADDL3	#38, 4(R2), MSG	1902
			0084	C6	9E	00061	MOVAB	132(R6), R8	1903
			1A	A2	3C	00066	MOVZWL	26(MSG), (R8)	
				1A	15	0006A	BLEQ	5\$	1904
			0088	C6	9F	0006C	PUSHAB	136(R6)	1910
				58	DD	00070	PUSHL	R8	
		0000G		02	FB	00072	CALLS	#2, OPC\$GET_VM	
				50	D0	00077	MOVL	R0, STATUS	
				59	E9	0007A	BLBC	STATUS, 12\$	
0088	D6	1C		68	28	0007D	MOVC3	(R8), 28(MSG), @136(R6)	1916
				0B	11	00084	BRB	6\$	1904
		50		8F	D0	00086	MOVL	#360572, MESSAGE_VECTOR	1923
			54	AE	D4	0008E	CLRL	MESSAGE_VECTOR+4	1924
			53	A6	91	00091	CMPB	83(R6), #4	1933
				05	12	00095	BNEQ	7\$	
		53		01	D0	00097	MOVL	#1, SCOPE_LIMIT	1935
				04	11	0009A	BRB	8\$	
		53	53	A6	9A	0009C	MOVZBL	83(R6), SCOPE_LIMIT	1937
				50	D4	000A0	CLRL	FOUND	1938
		52	53	A6	9A	000A2	MOVZBL	83(R6), SCOPE	1939
		53		52	D1	000A6	CMPL	SCOPE, SCOPE_LIMIT	1940
				16	19	000A9	BLSS	10\$	
		21		50	E8	000AB	BLBS	FOUND, 11\$	
			04	AE	9F	000AE	PUSHAB	OCD	1941
				67	DD	000B1	PUSHL	(R7)	
				52	DD	000B3	PUSHL	SCOPE	
0000G		CF		03	FB	000B5	CALLS	#3, FIND OCD	
		E9		50	E8	000BA	BLBS	FOUND, 9\$	
				52	D7	000BD	DECL	SCOPE	1943
				E5	11	000BF	BRB	9\$	1941
				50	E8	000C1	BLBS	FOUND, 11\$	1944
50		OB		8F	D0	000C4	MOVL	#360545, MESSAGE_VECTOR	1947
		AE	00058061						

		54	AE	D4	000CC	CLRL	MESSAGE_VECTOR+4	1948	
		50	AE	D5	000CF	TSTL	MESSAGE_VECTOR	1954	
		17	13	000D2	BEQL	13\$			
		50	AE	9F	000D4	PUSHAB	MESSAGE_VECTOR	1957	
		56	DD	000D7	PUSHL	R6			
0000G	CF	02	FB	000D9	CALLS	#2, FORMAT_MESSAGE			
		50	AE	9F	000DE	PUSHAB	MESSAGE_VECTOR	1958	
		56	DD	000E1	PUSHL	R6			
0000G	CF	02	FB	000E3	CALLS	#2, SEND_REPLY			
		00B2	31	000E8	BRW	18\$		1959	
	50	04	AE	D0	000EB	MOVL	OCD, R0	1965	
24	A6	50	D0	000EF	MOVL	R0, 36(R6)			
53	A6	0B	A0	90	000F3	MOVB	11(R0), 83(R6)	1966	
		6A	B5	000F8	TSTW	LCL_NODENAME		1970	
		26	13	000FA	BEQL	14\$			
	7E	01	7D	000FC	MOVQ	#1, -(SP)		1973	
		10	AE	9F	000FF	PUSHAB	ON_DSC		
		14	AE	9F	00102	PUSHAB	ON_DSC		
		000582AB	8F	DD	00105	PUSHL	#361131		
00000000G	00	05	FB	0010B	CALLS	#5, SYSSGETMSG			
	59	50	D0	00112	MOVL	R0, STATUS			
	OD	59	E8	00115	BLBS	STATUS, 15\$			
00000000G	00	59	DD	00118	PUSHL	STATUS		1975	
		01	FB	0011A	CALLS	#1, LIB\$STOP			
		04	00121	RET					
		08	AE	D4	00122	CLRL	ON_DSC	1978	
		64	AE	9F	00125	PUSHAB	MESSAGE_VECTOR+20	1982	
		68	DD	00128	PUSHL	(R8)			
		0088	C6	DD	0012A	PUSHL	136(R6)		
0000V	CF	03	FB	0012E	CALLS	#3, FORMAT_ALARM			
	59	50	D0	00133	MOVL	R0, STATUS			
1B	34	54	AE	D4	00136	CLRL	MESSAGE_VECTOR+4	1987	
		06	E1	00139	BBC	#6, 52(R6), 16\$		1988	
		59	E9	0013E	BLBC	STATUS, 16\$			
	50	8F	D0	00141	MOVL	#360971, MESSAGE_VECTOR		1991	
	58	AE	0005820B	08	AE	9E	00149	1992	
	5C	AE	6A	3C	0014E	MOVZWL	LCC_NODENAME, MESSAGE_VECTOR+12	1993	
	60	AE	AA	D0	00152	MOVL	LCL_NODENAME+4, MESSAGE_VECTOR+16	1994	
		20	11	00157	BRB	17\$		1988	
	50	AE	00058213	8F	D0	00159	16\$:	1998	
	58	AE	74	A6	3C	00161	MOVZWL	116(R6), MESSAGE_VECTOR+8	1999
	5C	AE	3C	A6	9E	00166	MOVAB	60(R6), MESSAGE_VECTOR+12	2000
	60	AE	08	AE	9E	0016B	MOVAB	ON_DSC, MESSAGE_VECTOR+16	2001
	64	AE	6A	3C	00170	MOVZWL	LCC_NODENAME, MESSAGE_VECTOR+20	2002	
	68	AE	04	AA	D0	00174	MOVL	LCL_NODENAME+4, MESSAGE_VECTOR+24	2003
		50	AE	9F	00179	17\$:	PUSHAB	MESSAGE_VECTOR	2006
		56	DD	0017C	PUSHL	R6			
0000G	CF	02	FB	0017E	CALLS	#2, FORMAT_MESSAGE		2010	
		56	DD	00183	PUSHL	R6			
		12	DD	00185	PUSHL	#18			
	7E	01	CE	00187	MNEGL	#1, -(SP)			
0000G	CF	03	FB	0018A	CALLS	#3, CLUSMSG_RQCB_SEND		2011	
		56	DD	0018F	PUSHL	R6			
0000G	CF	01	FB	00191	CALLS	#1, NOTIFY_LISTED_OPERATORS		2012	
		56	DD	00196	PUSHL	R6			
0000G	CF	01	FB	00198	CALLS	#1, LOG_MESSAGE		2014	
		56	DD	0019D	18\$:	PUSHL	R6		

OPCSSECURITY
V04-000

K 14
16-Sep-1984 01:48:50 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:50:55 [OPCOM.SRC]SECURITY.B32;1

Page 20
(3)

0000G CF

01 FB 0019F
04 0C1A4

CALLS #1, DEALLOCATE_RQCB
RET

: 2016

; Routine Size: 421 bytes, Routine Base: \$CODE\$ + 0000

```
721 2017 1 GLOBAL ROUTINE FORMAT_ALARM(RECORD_PTR, RECORD_SIZ, ALARM_DESCR_PTR) =
722 2018 1
723 2019 1 ++
724 2020 1 FUNCTIONAL DESCRIPTION:
725 2021 1
726 2022 1 This is the main formatting routine. It validates the security alarm
727 2023 1 record, and then dispatches by record type to routines which format
728 2024 1 the alarm message.
729 2025 1
730 2026 1 INPUT:
731 2027 1 RECORD_PTR : The security alarm record address.
732 2028 1 RECORD_SIZ : Size of the security alarm record
733 2029 1
734 2030 1 IMPLICIT INPUT:
735 2031 1 NONE.
736 2032 1
737 2033 1 OUTPUT:
738 2034 1 ALARM_DESCR_PTR : The address of the security alarm message descriptor
739 2035 1 is returned in this address.
740 2036 1
741 2037 1 IMPLICIT OUTPUT:
742 2038 1 NONE.
743 2039 1
744 2040 1 SIDE EFFECTS:
745 2041 1 NONE.
746 2042 1
747 2043 1 ROUTINE VALUE:
748 2044 1 $$$_NORMAL, a $FAO system service error, or 0 indicating the record is
749 2045 1 invalid.
750 2046 1 --
751 2047 1
752 2048 2 BEGIN ! Start of FORMAT_ALARM
753 2049 2
754 2050 2 MAP
755 2051 2 RECORD_PTR : REF $BLOCK;
756 2052 2
757 2053 2 LOCAL
758 2054 2 STATUS : LONG;
759 2055 2
760 2056 2 |
761 2057 2 | Validate the record
762 2058 2
763 2059 2 STATUS = VALIDATE_RECORD(.RECORD_PTR, .RECORD_SIZ);
764 2060 2 IF NOT .STATUS THEN RETURN .STATUS;
765 2061 2
766 2062 2 | Initialize the alarm descriptor
767 2063 2
768 2064 2 ALARM_DESCR[DSC$W_LENGTH] = OPC$K_MAXMESSAGE;
769 2065 2 ALARM_DESCR[DSC$A_POINTER] = ALARM_BUF;
770 2066 2
771 2067 2 | Dispatch by record type to the routines which format the messages
772 2068 2
773 2069 3 STATUS = (CASE .RECORD_PTR[NSA$W_REC_TYPE] FROM 1 TO NSA$K_RECTYPNUM OF
774 2070 3 SET
775 2071 3 [NSA$K_RECTYP_FILE] : FORMAT_FILE(.RECORD_PTR);
776 2072 3 [NSA$K_RECTYP_SYSUAF] : FORMAT_SYSUAF(.RECORD_PTR);
777 2073 3 [NSA$K_RECTYP_NETUAF] : FORMAT_NETUAF(.RECORD_PTR);
```



```
: 778      2074 3      [NSASK_RECTYP_LOGB]      : FORMAT_LOGB(.RECORD_PTR);
: 779      2075 3      [NSASK_RECTYP_LOGI]      : FORMAT_LOGI(.RECORD_PTR);
: 780      2076 3      [NSASK_RECTYP_LOGF]      : FORMAT_LOGF(.RECORD_PTR);
: 781      2077 3      [NSASK_RECTYP_LOGO]      : FORMAT_LOGO(.RECORD_PTR);
: 782      2078 3      [NSASK_RECTYP_VOL]       : FORMAT_VOL(.RECORD_PTR);
: 783      2079 2      TES);
: 784      2080 2      IF NOT .STATUS THEN RETURN .STATUS;
: 785      2081 2
: 786      2082 2      The alarm descriptor is modified as message lines are put in the buffer so
: 787      2083 2      we must restore it before we return.
: 788      2084 2
: 789      2085 2      ALARM_DESCR[DSC$W_LENGTH] = OPC$K_MAXMESSAGE - .ALARM_DESCR[DSC$W_LENGTH];
: 790      2086 2      ALARM_DESCR[DSC$A_POINTER] = ALARM_BUF;
: 791      2087 2
: 792      2088 2      Return the message descriptor.
: 793      2089 2
: 794      2090 2      .ALARM_DESCR_PTR = ALARM_DESCR;
: 795      2091 2      RETURN SSS_NORMAL;
: 796      2092 1      END;
```

! End of FORMAT_ALARM

				000C 00000	.ENTRY	FORMAT_ALARM, Save R2,R3	: 2017
	53	0000'	CF	9E 00002	MOVAB	ALARM_DESCR, R3	
		08	AC	DD 00007	PUSHL	RECORD_SIZ	: 2059
	52	04	AC	DD 0000A	MOVL	RECORD_PTR, R2	
			52	DD 0000E	PUSHL	R2	
	0000V	CF	02	FB 00010	CALLS	#2, VALIDATE_RECORD	
		7B	50	E9 00015	BLBC	STATUS, 11\$	: 2060
		63	0800	8F B0 00018	MOVW	#2048, ALARM_DESCR	: 2064
	04	A3	F800	C3 9E 0001D	MOVAB	ALARM_BUF, ALARM_DESCR+4	: 2065
		01		62 AF 00023	CASEW	(R2), #1, #7	: 2069
002B	07	0019	0010	00027 1\$:	.WORD	2\$-1\$,-	
004F	0022	003D	0034	0002F		3\$-1\$,-	
						4\$-1\$,-	
						5\$-1\$,-	
						6\$-1\$,-	
						7\$-1\$,-	
						8\$-1\$,-	
						9\$-1\$	
	0000V	CF	52	DD 00037 2\$:	PUSHL	R2	: 2071
			01	FB 00039	CALLS	#1, FORMAT_FIL	
			3D	11 0003E	BRB	10\$	
	0000V	CF	52	DD 00040 3\$:	PUSHL	R2	: 2072
			01	FB 00042	CALLS	#1, FORMAT_SYSUAF	
			34	11 00047	BRB	10\$	
	0000V	CF	52	DD 00049 4\$:	PUSHL	R2	: 2073
			01	FB 0004B	CALLS	#1, FORMAT_NETUAF	
			2B	11 00050	BRB	10\$	
	0000V	CF	52	DD 00052 5\$:	PUSHL	R2	: 2074
			01	FB 00054	CALLS	#1, FORMAT_LOGB	
			22	11 00059	BRB	10\$	
	0000V	CF	52	DD 0005B 6\$:	PUSHL	R2	: 2075
			01	FB 0005D	CALLS	#1, FORMAT_LOGI	
			19	11 00062	BRB	10\$	

OPCS\$SECURITY
V04-000

N 14
16-Sep-1984 01:48:50
14-Sep-1984 12:50:55

VAX-11 Bliss-32 V4.0-742
[OPCOM.SRC]SECURITY.B32;1

Page 23
(4)

	0000V	CF		52	DD	00064	7\$:	PUSHL	R2		:	2076
				01	FB	00066		CALLS	#1,	FORMAT_LOGF	:	
				10	11	00068		BRB	10\$		:	
	0000V	CF		52	DD	0006D	8\$:	PUSHL	R2		:	2077
				01	FB	0006F		CALLS	#1,	FORMAT_LOGO	:	
				07	11	00074		BRB	10\$		:	
	0000V	CF		52	DD	00076	9\$:	PUSHL	R2		:	2078
				01	FB	00078		CALLS	#1,	FORMAT_VOL	:	
63	0800	8F		50	E9	0007D	10\$:	BLBC	STATUS, 11\$		:	2080
	04	A3	F800	63	A3	00080		SUBW3	ALARM_DESCR, #2048, ALARM_DESCR		:	2085
	0C	BC		63	9E	00086		MOVAB	ALARM_BUF, ALARM_DESCR+4		:	2086
		50		63	9E	0008C		MOVAB	ALARM_DESCR, @ALARM_DESCR_PTR		:	2090
				01	D0	00090		MOVL	#1, R0		:	2091
				04	00093	11\$:		RET			:	2092

; Routine Size: 148 bytes, Routine Base: \$CODE\$ + 01A5


```

798 2093 1 ROUTINE VALIDATE_RECORD(RECORD_PTR, RECORD_SIZ) =
799 2094 1
800 2095 1 ++
801 2096 1 FUNCTIONAL DESCRIPTION:
802 2097 1
803 2098 1 This routine validates the security alarm record.
804 2099 1
805 2100 1 INPUT:
806 2101 1 RECORD_PTR : Security alarm record address.
807 2102 1 RECORD_SIZ : Size of the record.
808 2103 1
809 2104 1 IMPLICIT INPUT:
810 2105 1 NONE.
811 2106 1
812 2107 1 OUTPUT:
813 2108 1 NONE.
814 2109 1
815 2110 1 IMPLICIT OUTPUT:
816 2111 1 NONE.
817 2112 1
818 2113 1 SIDE EFFECTS:
819 2114 1 NONE.
820 2115 1
821 2116 1 ROUTINE VALUE:
822 2117 1 SSS_NORMAL or 0 if the record is not valid.
823 2118 1 --
824 2119 1
825 2120 2 BEGIN ! Start of VALIDATE_RECORD
826 2121 2
827 2122 2 MAP
828 2123 2 RECORD_PTR : REF $BLOCK;
829 2124 2
830 2125 2 LOCAL
831 2126 2 PKT_PTR : REF $BLOCK,
832 2127 2 SUBTYPE_PTR : REF VECTOR,
833 2128 2 RECID_PTR : REF VECTOR[.BYTE],
834 2129 2 PKT_TYPE : LONG,
835 2130 2 PKT_SIZE : LONG,
836 2131 2 INDEX : LONG INITIAL(0),
837 2132 2 COUNT : LONG INITIAL(0),
838 2133 2 I : LONG;
839 2134 2
840 2135 2 !
841 2136 2 Validate the record header fields.
842 2137 2
843 2138 2 IF (.RECORD_SIZ LSSU NSASK_RECHDR_LENGTH) OR
844 2139 2 (.RECORD_PTR[NSASW_REC_TYPE] EQLU 0) OR
845 2140 2 (.RECORD_PTR[NSASW_REC_TYPE] GTRU .RECTYP[0]) OR
846 2141 2 (.RECORD_PTR[NSASW_REC_SUBTYPE] EQLU 0) OR
847 2142 2 (.RECORD_PTR[NSASW_REC_SUBTYPE] GTRU .RECTYP[.RECORD_PTR[NSASW_REC_TYPE]]) OR
848 2143 2 (.RECORD_PTR[NSASB_REC_SEQNUM] NEQU 1) OR
849 2144 2 (.RECORD_PTR[NSASB_REC_SEQLAST] NEQU 1) OR
850 2145 2 (.RECORD_PTR[NSASB_REC_FLAGS] NEQU 0) OR
851 2146 2 (.RECORD_PTR[NSASW_REC_PKTOFF]-NSASK_PKTHDR_LENGTH NEQU NSASK_RECHDR_LENGTH) OR
852 2147 2 (.RECORD_PTR[NSASW_REC_PKTHDRSIZ] NEQU NSASK_PKTHDR_LENGTH)
853 2148 2 THEN
854 2149 2 RETURN 0;
```



```
855 2150 2 | Initialize the packet count table and get a pointer to the first packet.
856 2151 2 |
857 2152 2 | CH$FILL(0, NSASK_PKTTPNUM, PKT_COUNT_TAB);
858 2153 2 | PKT_PTR = .RECORD_PTR + .RECORD_PTR[NSASW_REC_PKTOFF];
859 2154 2 |
860 2155 2 | If the packet count is not zero, we must perform various validation checks
861 2156 2 | and set up the packet index and packet pointer tables.
862 2157 2 |
863 2158 2 | IF .RECORD_PTR[NSASB_REC_PKTNUM] GTRU 0
864 2159 2 | THEN
865 2160 2 | BEGIN
866 2161 3 |   DECR I FROM .RECORD_PTR[NSASB_REC_PKTNUM] TO 1 DO
867 2162 3 |   BEGIN
868 2163 4 |     Get the packet size and type.
869 2164 4 |
870 2165 4 |     PKT_TYPE = .PKT_PTR[NSASW_PKT_TYPE];
871 2166 4 |     PKT_SIZE = .PKT_PTR[NSASW_PKT_SIZE];
872 2167 4 |
873 2168 4 |     Validate the packet size and type.
874 2169 4 |
875 2170 4 |     IF (.PKT_TYPE EQLU 0) OR
876 2171 4 |       (.PKT_TYPE GTRU NSASK_PKTTPNUM) OR
877 2172 4 |       (.PKT_SIZE - NSASK_PKTHDR_LENGTH LSSU .PKT_SIZE_TAB[.PKT_TYPE-1, 0]) OR
878 2173 4 |       (.PKT_SIZE - NSASK_PKTHDR_LENGTH GTRU .PKT_SIZE_TAB[.PKT_TYPE-1, 1])
879 2174 4 |     THEN
880 2175 5 |       RETURN 0;
881 2176 4 |
882 2177 4 |     Increment the packet count, get the next packet address and
883 2178 4 |     make sure that it does not point past the end of the record.
884 2179 4 |
885 2180 4 |     PKT_COUNT_TAB[.PKT_TYPE-1] = .PKT_COUNT_TAB[.PKT_TYPE-1] + 1;
886 2181 4 |     PKT_PTR = .PKT_PTR + .PKT_SIZE;
887 2182 4 |     IF (.I NEQU 1) AND
888 2183 4 |       (.PKT_PTR - .RECORD_PTR - NSASK_PKTHDR_LENGTH GEQU .RECORD_SIZ)
889 2184 4 |     THEN
890 2185 5 |       RETURN 0;
891 2186 4 |
892 2187 4 |     END;
893 2188 3 |
894 2189 3 | The packet data types and sizes have been validated. The next step
895 2190 3 | is to set up the packet index and pointer tables. First we set up
896 2191 3 | the packet index table.
897 2192 3 |
898 2193 3 | INCR I FROM 0 TO NSASK_PKTTPNUM-1 DO
899 2194 3 | BEGIN
900 2195 4 |   IF .PKT_COUNT_TAB[.I] NEQU 0
901 2196 4 |   THEN
902 2197 4 |   BEGIN
903 2198 5 |     PKT_INDEX_TAB[.I] = .INDEX;
904 2199 5 |     INDEX = .INDEX + .PKT_COUNT_TAB[.I];
905 2200 5 |   END;
906 2201 4 |   END;
907 2202 3 | END;
908 2203 3 |
909 2204 3 | Now we step through the packets one more time to set up the
910 2205 3 | packet pointer table. In the process we re-count the packets.
911 2206 3 |
```



```

: 912      2207 3 CH$FILL(0, NSASK_PKTTPNUM, PKT_COUNT_TAB);
: 913      2208 3 PKT_PTR = .RECORD_PTR + .RECORD_PTR[NSASK_REC_PKTOFF];
: 914      2209 3 DECR I FROM .RECORD_PTR[NSASK_REC_PKTNUM] TO 1 DO
: 915      2210 4 BEGIN
: 916      2211 4 |
: 917      2212 4 |   Get the packet type, save the packet address in the packet
: 918      2213 4 |   pointer table, increment the packet count, and get the next
: 919      2214 4 |   packet pointer.
: 920      2215 4 |
: 921      2216 4 |   PKT_TYPE = .PKT_PTR[NSASK_PKT_TYPE];
: 922      2217 4 |   PKT_POINTER_TAB[.PKT_INDEX_TAB[.PKT_TYPE-1]+.PKT_COUNT_TAB[.PKT_TYPE-1]] = .PKT_PTR;
: 923      2218 4 |   PKT_COUNT_TAB[.PKT_TYPE-1] = .PKT_COUNT_TAB[.PKT_TYPE-1] + 1;
: 924      2219 4 |   PKT_PTR = .PKT_PTR + .PKT_PTR[NSASK_PKT_SIZE];
: 925      2220 3 END;
: 926      2221 3 END;
: 927      2222 2 |
: 928      2223 2 |   Now we check that the right packets are present for this type of record.
: 929      2224 2 |
: 930      2225 2 |   First we form a pointer to the table which describes the packet types
: 931      2226 2 |   for this record.
: 932      2227 2 |
: 933      2228 2 |   SUBTYPE_PTR = .RECTYPE[RECORD_PTR[NSASK_REC_TYPE]];
: 934      2229 2 |   RECID_PTR = .SUBTYPE_PTR[RECORD_PTR[NSASK_REC_SUBTYPE]];
: 935      2230 2 |
: 936      2231 2 |   Then we make sure that each packet found is allowed in this record.
: 937      2232 2 |
: 938      2233 2 |   INCR I FROM 1 TO .RECID_PTR[0]*3-2 BY 3 DO
: 939      2234 3 BEGIN
: 940      2235 3 |   IF (.PKT_COUNT_TAB[.RECID_PTR[I]-1] LSSU .RECID_PTR[I+1]) OR
: 941      2236 3 |   (.PKT_COUNT_TAB[.RECID_PTR[I]-1] GTRU .RECID_PTR[I+2])
: 942      2237 4 THEN
: 943      2238 3 |   RETURN 0
: 944      2239 3 |   ELSE
: 945      2240 3 |   COUNT = .COUNT + .PKT_COUNT_TAB[.RECID_PTR[I]-1];
: 946      2241 2 |   END;
: 947      2242 2 |
: 948      2243 2 |   The packet count can now be used to make sure that there were not any
: 949      2244 2 |   additional packet types in the record.
: 950      2245 2 |
: 951      2246 2 |   IF .COUNT NEQU .RECORD_PTR[NSASK_REC_PKTNUM]
: 952      2247 2 |   THEN
: 953      2248 2 |   RETURN 0;
: 954      2249 2 |
: 955      2250 2 |   The last check is to validate the record size.
: 956      2251 2 |
: 957      2252 2 |   IF .PKT_PTR - .RECORD_PTR - NSASK_PKT_HDR_LENGTH NEQU .RECORD_SIZE
: 958      2253 2 |   THEN
: 959      2254 2 |   RETURN 0;
: 960      2255 2 |
: 961      2256 2 |   RETURN SS$NORMAL;
: 962      2257 2 |
: 963      2258 1 END;

```

! End of VALIDATE_RECORD

OFFC 00000 VALIDATE_RECORD:

				7E D4 00002	WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	
				5B D4 00004	CLRL	INDEX	2093
				08 AC D1 00006	CLRL	COUNT	2120
	0000004C	8F		03 1E 0000E	CMPL	RECORD_SIZ, #76	2138
				017B 31 00010	BGEQU	2\$	
				04 BC B5 00013	BRW	15\$	
				F8 13 00016	TSTW	@RECORD_PTR	2139
0000'	CF	04	BC	00 ED 00018	BEQL	1\$	
		10		EE 1A 00020	CMPZV	#0, #16, @RECORD_PTR, RECTYP	2140
		50		04 AC D0 00022	BGTRU	1\$	
				02 A0 B5 00026	MOVL	RECORD_PTR, R0	2141
				E5 13 00029	TSTW	2(R0)	
		51		04 AC D0 0002B	BEQL	1\$	
		50		61 3C 0002F	MOVL	RECORD_PTR, R1	2142
		50		0000'CF40 D0 00032	MOVZWL	(R1), R0	
60		02	A1	00 ED 00038	MOVL	RECTYP[R0], R0	
		10		D0 1A 0003E	CMPZV	#0, #16, 2(R1), (R0)	
		50		04 AC D0 00040	BGTRU	1\$	
		01		04 A0 91 00044	MOVL	RECORD_PTR, R0	2143
				C6 12 00048	CMPB	4(R0), #1	
		50		04 AC D0 0004A	BNEQ	1\$	
		01		05 A0 91 0004E	MOVL	RECORD_PTR, R0	2144
				BC 12 00052	CMPB	5(R0), #1	
		50		04 AC D0 00054	BNEQ	1\$	
				06 A0 95 00058	MOVL	RECORD_PTR, R0	2145
				B3 12 0005B	TSTB	6(R0)	
		50		04 AC D0 0005D	BNEQ	1\$	
	0050	8F		08 A0 B1 00061	MOVL	RECORD_PTR, R0	2146
				A7 12 00067	CMPW	8(R0), #80	
		50		04 AC D0 00069	BNEQ	1\$	
		04		0A A0 B1 0006D	MOVL	RECORD_PTR, R0	2147
				9D 12 00071	CMPW	10(R0), #4	
13		00		00 2C 00073	BNEQ	1\$	
		6E		0000'CF 00078	MOVCS	#0, (SP), #0, #19, PKT_COUNT_TAB	2153
		58		04 AC D0 0007B			2154
		59		08 A8 3 0007F	MOVL	RECORD_PTR, R8	
		59		58 C 00083	MOVZWL	8(R8), R9	
		56		59 D0 00086	ADDL2	R8, R9	
		5A		07 A8 9A 00089	MOVL	R9, PKT_PTR	
				03 12 0008D	MOVZBL	7(R8), R10	2159
				00A7 31 0008F	BNEQ	3\$	
		50		01 AA 9E 00092	BRW	12\$	
				4E 11 00096	MOVAB	1(R10), I	2162
		57		FC A6 3C 00098	BRB	7\$	
		51		FE A6 3C 0009C	MOVZWL	-4(PKT_PTR), PKT_TYPE	2167
				57 D5 000A0	MOVZWL	-2(PKT_PTR), PKT_SIZE	2168
				25 13 000A2	TSTL	PKT_TYPE	2172
		13		57 D1 000A4	BEQL	5\$	
				20 1A 000A7	CMPL	PKT_TYPE, #19	2173
		52		FC A1 9E 000A9	BGTRU	5\$	
				0000'CF47 DF 000AD	MOVAB	-4(R1), R2	2174
		10		00 ED 000B2	PUSHAL	PKT_SIZE_TAB-4[PKT_TYPE]	
52				10 1A 000B7	CMPZV	#0, #16, @ (SP)+, R2	
		52		FC A1 9E 000B9	BGTRU	5\$	
				0000'CF47 DF 000BD	MOVAB	-4(R1), R2	2175
					PUSHAL	PKT_SIZE_TAB-2[PKT_TYPE]	

52	9E	10	00	ED	000C2	CMPZV	#0, #16, @ (SP)+, R2	:		
			03	1E	000C7	BGEQU	6\$	:		
			00C2	31	000C9	BRW	15\$	:		
			0000'CF47	96	000CC	INCB	PKT_COUNT_TAB-1[PKT_TYPE]	:	2182	
	56		51	C0	000D1	ADDL2	PKT_SIZE, -PKT_PTR	:	2183	
	01		50	D1	000D4	CMPL	I, #1	:	2184	
			0D	13	000D7	BEQL	7\$	:		
	52	56	58	C3	000D9	SUBL3	R8, PKT_PTR, R2	:	2185	
		52	04	C2	000DD	SUBL2	#4, R2	:		
	08	AC	52	D1	000E0	CMPL	R2, RECORD_SIZ	:		
			E3	1E	000E4	BGEQU	5\$	:		
	AF		50	F5	000E6	SOBGR	I, 4\$	:	2162	
			50	D4	000E9	CLRL	I	:	2194	
		51	0000'CF40	9A	000EB	MOVZBL	PKT_COUNT_TAB[I], R1	:	2196	
			09	13	000F1	BEQL	9\$	:		
		0000'CF40	6E	B0	000F3	MOVW	INDEX, PKT_INDEX_TAB[I]	:	2199	
		6E	51	C0	000F9	ADDL2	R1, INDEX	:	2200	
	13	EB	50	12	F3	AOBLEQ	#18, I, 8\$	:	2194	
	00	6E	00	2C	00100	MOVC5	#0, (SP), #0, #19, PKT_COUNT_TAB	:	2207	
			0000'	CF	00105			:		
		56	59	D0	00108	MOVL	R9, PKT_PTR	:	2208	
		50	01	AA	0010B	MOVAB	1(R10), -I	:	2209	
			25	11	0010F	BRB	11\$	:		
		57	FC	A6	3C	00111	MOVZWL	-4(PKT_PTR), PKT_TYPE	:	2216
		51	0000'CF47	3C	00115	MOVZWL	PKT_INDEX_TAB-2[PKT_TYPE], R1	:	2217	
		52	0000'CF47	9A	0011B	MOVZBL	PKT_COUNT_TAB-1[PKT_TYPE], R2	:		
		51	52	C0	00121	ADDL2	R2, R1	:		
		0000'CF41	56	D0	00124	MOVL	PKT_PTR, PKT_POINTER_TAB[R1]	:		
			0000'CF47	96	0012A	INCB	PKT_COUNT_TAB-1[PKT_TYPE]	:	2218	
		51	FE	A6	3C	0012F	MOVZWL	-2(PKT_PTR), R1	:	2219
		56	51	C0	00133	ADDL2	R1, PKT_PTR	:		
		D8	50	F5	00136	SOBGR	I, 10\$	:	2209	
		50	68	3C	00139	MOVZWL	(R8), R0	:	2228	
		51	0000'CF40	D0	0013C	MOVL	RECTYPE[R0], SUBTYPE_PTR	:		
		50	02	A8	3C	00142	MOVZWL	2(R8), R0	:	2229
		50	6140	D0	00146	MOVL	(SUBTYPE_PTR)[R0], RECID_PTR	:		
		51	60	9A	0014A	MOVZBL	(RECID_PTR), R1	:	2233	
		51	03	C4	0014D	MULL2	#3, R1	:		
		51	02	C2	00150	SUBL2	#2, R1	:		
		52	02	CE	00153	MNEGL	#2, I	:		
			1B	11	00156	BRB	14\$	:		
		53	6240	9A	00158	MOVZBL	(I)[RECID_PTR], R3	:	2235	
		53	0000'CF43	9A	0015C	MOVZBL	PKT_COUNT_TAB-1[R3], R3	:		
		53	01 A240	91	00162	CMPB	1(I)[RECID_PTR], R3	:		
			25	1A	00167	BGTRU	15\$	:		
		53	02 A240	91	00169	CMPB	2(I)[RECID_PTR], R3	:	2236	
			1E	1F	0016E	BLSSU	15\$	:		
	FFDF	52	5B	53	C0	00170	ADDL2	R3, COUNT	:	2240
		03	51	F1	00173	ACBL	R1, #3, I, 13\$	:	2233	
		5A	5B	D1	00179	CMPL	COUNT, R10	:	2246	
			10	12	0017C	BNEQ	15\$	:		
		56	58	C2	0017E	SUBL2	R8, R6	:	2253	
		56	04	C2	00181	SUBL2	#4, R6	:		
		08	AC	56	D1	00184	CMPL	R6, RECORD_SIZ	:	
			04	12	00188	BNEQ	15\$	:		
		50	01	D0	0018A	MOVL	#1, R0	:	2257	
			04	0018D	RET			:		

OPCSSECURITY
V04-000

G 15
16-Sep-1984 01:48:50
14-Sep-1984 12:50:55

VAX-11 Bliss-32 V4.0-742
[OPCOM.SRC]SECURITY.B32;1

Page 29
(5)

50 D4 0018E 15\$: CLRL R0
04 00190 RET

; 2258
;

; Routine Size: 401 bytes, Routine Base: \$CODE\$ + 0239


```

965 2259 1 ROUTINE FORMAT_FIL(RECORD_PTR) =
966 2260 1
967 2261 1 ++
968 2262 1 FUNCTIONAL DESCRIPTION:
969 2263 1
970 2264 1 This routine formats security alarm messages for file access alarms.
971 2265 1
972 2266 1 INPUT:
973 2267 1 RECORD_PTR : The security alarm record address.
974 2268 1
975 2269 1 IMPLICIT INPUT:
976 2270 1 NONE.
977 2271 1
978 2272 1 OUTPUT:
979 2273 1 NONE.
980 2274 1
981 2275 1 IMPLICIT OUTPUT:
982 2276 1 Formatted message in the alarm buffer.
983 2277 1
984 2278 1 SIDE EFFECTS:
985 2279 1 NONE.
986 2280 1
987 2281 1 ROUTINE VALUE:
988 2282 1 SSS_NORMAL or a $FAO system service error.
989 2283 1 --
990 2284 1
991 2285 2 BEGIN ! Start of FORMAT_FIL
992 2286 2
993 2287 2 MAP
994 2288 2 RECORD_PTR : REF $BLOCK;
995 2289 2
996 2290 2 OWN
997 2291 2 FIL_VEC : VECTOR[NSASK_RECTYPNUM FIL]
998 2292 2 INITIAL(UPLIT(BYTE(%ASCIC ' / Successful file access')),
999 2293 2 UPLIT(BYTE(%ASCIC ' / File access failure')));
1000 2294 2
1001 2295 2 LOCAL
1002 2296 2 STATUS : LONG;
1003 2297 2
1004 2298 2
1005 2299 2 Format the common message lines.
1006 2300 2
1007 2301 2 STATUS = PUT_COMMON(.RECORD_PTR, .FIL_VEC[.RECORD_PTR[NSASK_REC_SUBTYPE]-1]);
1008 2302 2 IF NOT .STATUS THEN RETURN .STATUS;
1009 2303 2
1010 2304 2 Format the image name, file name, and file access mode lines.
1011 2305 2
1012 2306 2 PUT_PKT_IMGNAME(.RECORD_PTR);
1013 2307 2 PUT_PKT_FILNAME(.RECORD_PTR);
1014 2308 2 PUT_PKT_FACMOD(.RECORD_PTR);
1015 2309 2
1016 2310 2 If it was a successful file access, format the privileges used line.
1017 2311 2
1018 2312 2 IF .RECORD_PTR[NSASK_REC_SUBTYPE] EQLU NSASK_RECTYP_FIL_SUCC
1019 2313 2 THEN
1020 2314 2 PUT_PKT_PRIVUSED(.RECORD_PTR);
1021 2315 2 RETURN SSS_NORMAL;
```

; 1022

2316 1 END;

! End of FORMAT_FIL

```
.PSECT $SPLITS,NOWRT,NOEXE,2
20 6C 75 66 73 73 65 63 63 75 53 20 2F 20 19 0000F .BLKB 1
73 73 65 63 63 61 20 65 6C 69 66 00010 P.AAD: .ASCII <25>\ / Successful file access\
73 73 65 63 63 61 20 65 6C 69 46 20 2F 20 16 0001F .BLKB 2
65 72 75 6C 69 61 66 20 0002A P.AAE: .ASCII <22>\ / File access failure\
0002C
0003B

.PSECT $OWNS,NOEXE,2
00000000' 00000000' 00AF0 FIL_VEC:.ADDRESS P.AAD, P.AAE

.PSECT $CODE$,NOWRT,2
0004 00000 FORMAT_FIL:
52 04 AC D0 00002 .WORD Save R2
50 02 A2 3C 00006 .MOV RECORD_PTR, R2
0000'CF 40 DD 0000A .MOVZWL 2(R2), -R0
52 DD 0000F .PUSHL FIL_VEC-4[R0]
0000V CF 02 FB 00011 .PUSHL R2
25 50 E9 00016 .CALLS #2, PUT_COMMON
52 DD 00019 .BLBC STATUS, -2$
0000V CF 01 FB 0001B .PUSHL R2
52 DD 00020 .CALLS #1, PUT_PKT_IMGNAM
0000V CF 01 FB 00022 .PUSHL R2
52 DD 00027 .CALLS #1, PUT_PKT_FILNAM
0000V CF 01 FB 00029 .PUSHL R2
01 02 A2 B1 0002E .CALLS #1, PUT_PKT_FACMOD
07 12 00032 .CMPW 2(R2), #1
52 DD 00034 .BNEQ 1$
0000V CF 01 FB 00036 .PUSHL R2
50 01 D0 0003B 1$: .CALLS #1, PUT_PKT_PRIVUSED
04 0003E 2$: .MOV #1, R0
RET
```

; Routine Size: 63 bytes, Routine Base: \$CODE\$ + 03CA


```

: 1024 2317 1 ROUTINE FORMAT_SYSUAF(RECORD_PTR) =
: 1025 2318 1
: 1026 2319 1 !++
: 1027 2320 1 FUNCTIONAL DESCRIPTION:
: 1028 2321 1
: 1029 2322 1 This routine formats security alarm messages for records whose type
: 1030 2323 1 is NSASK_RECTYP_SYSUAF. (System user authorization file)
: 1031 2324 1
: 1032 2325 1 INPUT:
: 1033 2326 1 RECORD_PTR : The security alarm record address.
: 1034 2327 1
: 1035 2328 1 IMPLICIT INPUT:
: 1036 2329 1 NONE.
: 1037 2330 1
: 1038 2331 1 OUTPUT:
: 1039 2332 1 NONE.
: 1040 2333 1
: 1041 2334 1 IMPLICIT OUTPUT:
: 1042 2335 1 Formatted message in the alarm buffer.
: 1043 2336 1
: 1044 2337 1 SIDE EFFECTS:
: 1045 2338 1 NONE.
: 1046 2339 1
: 1047 2340 1 ROUTINE VALUE:
: 1048 2341 1 $$$_NORMAL or a $FA0 system service error.
: 1049 2342 1 --
: 1050 2343 1
: 1051 2344 2 BEGIN ! Start of FORMAT_SYSUAF
: 1052 2345 2
: 1053 2346 2 MAP
: 1054 2347 2 RECORD_PTR : REF $BLOCK;
: 1055 2348 2
: 1056 2349 2 BIND
: 1057 2350 2 FROMNAM_BUF = UPLIT(BYTE(%ASCIC ' From Name: ')),
: 1058 2351 2 OLDNAM_BUF = UPLIT(BYTE(%ASCIC ' Old Name: '));
: 1059 2352 2
: 1060 2353 2 OWN
: 1061 2354 2 BANNER_VEC : VECTOR[NSASK_RECTYPNUM_SYSUAF]
: 1062 2355 2 INITIAL(UPLIT(BYTE(%ASCIC ' / System UAF record addition')),
: 1063 2356 2 UPLIT(BYTE(%ASCIC ' / System UAF record deletion')),
: 1064 2357 2 UPLIT(BYTE(%ASCIC ' / System UAF record modification')),
: 1065 2358 2 UPLIT(BYTE(%ASCIC ' / System UAF record copied')),
: 1066 2359 2 UPLIT(BYTE(%ASCIC ' / System UAF record renamed'))),
: 1067 2360 2 LABEL_VEC : VECTOR[NSASK_RECTYPNUM_SYSUAF]
: 1068 2361 2 INITIAL(UPLIT(BYTE(%ASCIC ' Rec Add: ')),
: 1069 2362 2 UPLIT(BYTE(%ASCIC ' Rec Del: ')),
: 1070 2363 2 UPLIT(BYTE(%ASCIC ' Rec Mod: ')),
: 1071 2364 2 UPLIT(BYTE(%ASCIC ' Rec Add: ')),
: 1072 2365 2 UPLIT(BYTE(%ASCIC ' New Name: ')));
: 1073 2366 2
: 1074 2367 2 LOCAL
: 1075 2368 2 STATUS : LONG;
: 1076 2369 2
: 1077 2370 2 !
: 1078 2371 2 Format the common message lines.
: 1079 2372 2
: 1080 2373 2 STATUS = PUT_COMMON(.RECORD_PTR, .BANNER_VEC[.RECORD_PTR[NSA$W_REC_SUBTYPE]-1]);

```



```
: 1081      2374 2 IF NOT .STATUS THEN RETURN .STATUS;
: 1082      2375
: 1083      2376 2 Move the user name label and user name fields into the alarm buffer.
: 1084      2377
: 1085      2378 2 PUT_STRING ASCII(.LABEL_VEC[.RECORD_PTR[NSASW_REC_SUBTYPE]-1]);
: 1086      2379 2 PUT_PKT_STRING(.RECORD_PTR, NSASK_PRTTYP_USERNAM, 1);
: 1087      2380 2 PUT_NEWLINE(0);
: 1088      2381
: 1089      2382 2 If this is a UAF copied or renamed record, format the old user name.
: 1090      2383
: 1091      2384 2 IF .RECORD_PTR[NSASW_REC_SUBTYPE] EQLU NSASK_RECTYP_SYSUAF_COP
: 1092      2385 2 OR .RECORD_PTR[NSASW_REC_SUBTYPE] EQLU NSASK_RECTYP_SYSUAF_REN
: 1093      2386 2 THEN
: 1094      2387 2 BEGIN
: 1095      2388 2 PUT_STRING ASCII(
: 1096      2389 2 IF .RECORD_PTR[NSASW_REC_SUBTYPE] EQLU NSASK_RECTYP_SYSUAF_COP
: 1097      2390 2 THEN FROMNAM_BUF
: 1098      2391 2 ELSE OLDNAM_BUF
: 1099      2392 2 );
: 1100      2393 2 PUT_PKT_STRING(.RECORD_PTR, NSASK_PRTTYP_USERNAM, 2);
: 1101      2394 2 PUT_NEWLINE(0);
: 1102      2395 2 END;
: 1103      2396
: 1104      2397 2 If this is anything but a UAF deletion record, format the modified fields.
: 1105      2398 2 ***** This is temporarily tied off because AUTHORIZE is not generating
: 1106      2399 2 ***** the fields modified vector.
: 1107      2400
: 1108      2401 2 IF .RECORD_PTR[NSASW_REC_SUBTYPE] NEQU NSASK_RECTYP_SYSUAF_DEL
: 1109      2402 2 THEN
: 1110      2403 2 IF .(UPLIT BYTE (0)) THEN ! ***** temporary tie-off
: 1111      2404 2 PUT_PKT_SYSUAF(.RECORD_PTR);
: 1112      2405 2 RETURN $$$_NORMAL;
: 1113      2406 1 END; ! End of FORMAT_SYSUAF
```

```
.PSECT $SPLITS,NOWRT,NOEXE,2
00043 .BLKB 1
00044 P.AAF: .ASCII <12><9>\From Name:\<9>
00051 .BLKB 3
00054 P.AAG: .ASCII <11><9>\Old Name:\<9>
00060 P.AAH: .ASCII <29>\ / System UAF record addition\
0006F
0007E .BLKB 2
00080 P.AAI: .ASCII <29>\ / System UAF record deletion\
0008F
0009E .BLKB 2
000A0 P.AAJ: .ASCII \! / System UAF record modification\
000AF
000BE
000C2 .BLKB 2
000C4 P.AAK: .ASCII <27>\ / System UAF record copied\
000D3
000E0 P.AAL: .ASCII <28>\ / System UAF record renamed\
000EF
000FD .BLKB 3
```



```

09 3A 64 64 41 20 63 65 52 09 0A 00100 P.AAM: .ASCII <10><9>\Rec Add:\<9>
09 3A 6C 65 44 20 63 65 52 09 0A 0010B .BLKB 1
09 3A 6C 65 44 20 63 65 52 09 0A 0010C P.AAN: .ASCII <10><9>\Rec Del:\<9>
09 3A 64 6F 4D 20 63 65 52 09 0A 00117 .BLKB 1
09 3A 64 6F 4D 20 63 65 52 09 0A 00118 P.AAO: .ASCII <10><9>\Rec Mod:\<9>
09 3A 64 64 41 20 63 65 52 09 0A 00123 .BLKB 1
09 3A 64 64 41 20 63 65 52 09 0A 00124 P.AAP: .ASCII <10><9>\Rec Add:\<9>
09 3A 65 6D 61 4E 20 77 65 4E 09 0B 0012F .BLKB 1
09 3A 65 6D 61 4E 20 77 65 4E 09 0B 00130 P.AAQ: .ASCII <11><9>\New Name:\<9>
00 0013C P.AAR: .BYTE 0

```

.PSECT \$OWNS,NOEXE,2

```

00000000' 00000000' 00000000' 00000000' 00000000' 00AF8 BANNER_VEC:
00000000' 00000000' 00000000' 00000000' 00000000' 00B0C LABEL_VEC:

```

```

FROMNAM_BUF= P.AAF
OLDNAM_BUF= P.AAG

```

.PSECT \$CODE\$,NOWRT,2

```

0004 00000 FORMAT_SYSUAF:
52 04 AC D0 00002 .WORD Save R2
50 02 A2 3C 00006 MOVL RECORD_PTR, R2
0000'CF40 DD 0000A MOVZWL 2(R2), R0
52 DD 0000F PUSHL BANNER_VEC-4[R0]
0000V CF 02 FB 00011 PUSHL R2
6F 50 E9 00016 CALLS #2, PUT_COMMON
50 02 A2 3C 00019 BLBC STATUS, 7$
0000'CF40 DD 0001D MOVZWL 2(R2), R0
0000V CF 01 FB 00022 PUSHL LABEL_VEC-4[R0]
01 DD 00027 CALLS #1, PUT_STRING_ASCII
0A DD 00029 PUSHL #1
52 DD 0002B PUSHL #10
0000V CF 03 FB 0002D CALLS #3, PUT_PKT_STRING
7E D4 00032 CLRL -(SP)
0000V CF 01 FB 00034 CALLS #1, PUT_NEWLINE
50 D4 00039 CLRL R0
04 02 A2 B1 0003B CMPW 2(R2), #4
04 12 0003F BNEQ 1$
50 D6 00041 INCL R0
06 11 00043 BRB 2$
05 02 A2 B1 00045 1$: CMPW 2(R2), #5
28 12 00049 BNEQ 5$
07 50 E9 0004B 2$: BLBC R0, 3$
50 0000'CF 9E 0004E MOVAB FROMNAM_BUF, R0
05 11 00053 BRB 4$
50 0000'CF 9E 00055 3$: MOVAB OLDNAM_BUF, R0
01 DD 0005A 4$: PUSHL R0
0000V CF 01 FB 0005C CALLS #1, PUT_STRING_ASCII
02 DD 00061 PUSHL #2
0A DD 00063 PUSHL #10
52 DD 00065 PUSHL R2

```

6
7

OPC\$SECURITY
V04-000

M 15
16-Sep-1984 01:48:50
14-Sep-1984 12:50:55

VAX-11 Bliss-32 V4.0-742
[OPCOM.SRC]SECURITY.B32;1

Page 35
(7)

0000V	CF		03	FB	00067	CALLS	#3, PUT_PKT_STRING	:	
			7E	D4	0006C	CLRL	-(SP)	:	2394
0000V	CF		01	FB	0006E	CALLS	#1, PUT_NEWLINE	:	
	02	02	A2	B1	00073	CMPW	2(R2), #2	:	2401
			0C	13	00077	BEQL	6\$	:	
	07	0000'	CF	E9	00079	BLBC	P.AAR, 6\$	:	2403
			52	DD	0007E	PUSHL	R2	:	2404
0000V	CF		01	FB	00080	CALLS	#1, PUT_PKT_SYSUAFF	:	
	50		01	D0	00085	MOVL	#1, R0	:	2405
			04	00088	7\$:	RET		:	2406

; Routine Size: 137 bytes, Routine Base: \$CODE\$ + 0409


```
: 1115      2407 1 ROUTINE FORMAT_NETUAF(RECORD_PTR) =
: 1116      2408 1
: 1117      2409 1 ++
: 1118      2410 1 FUNCTIONAL DESCRIPTION:
: 1119      2411 1
: 1120      2412 1     This routine formats security alarm messages for records whose type
: 1121      2413 1     is NSASK_RECTYP_NETUAF. (Network user authorization file)
: 1122      2414 1
: 1123      2415 1 INPUT:
: 1124      2416 1     RECORD_PTR      : The security alarm record address.
: 1125      2417 1
: 1126      2418 1 IMPLICIT INPUT:
: 1127      2419 1     NONE.
: 1128      2420 1
: 1129      2421 1 OUTPUT:
: 1130      2422 1     NONE.
: 1131      2423 1
: 1132      2424 1 IMPLICIT OUTPUT:
: 1133      2425 1     Formatted message in the alarm buffer.
: 1134      2426 1
: 1135      2427 1 SIDE EFFECTS:
: 1136      2428 1     NONE.
: 1137      2429 1
: 1138      2430 1 ROUTINE VALUE:
: 1139      2431 1     SSS_NORMAL or a $FAO system service error.
: 1140      2432 1 --
: 1141      2433 1
: 1142      2434 2 BEGIN                                ! Start of FORMAT_NETUAF
: 1143      2435 2
: 1144      2436 2 MAP
: 1145      2437 2     RECORD_PTR : REF $BBLOCK;
: 1146      2438 2
: 1147      2439 2 BIND
: 1148      2440 2     OLDREC_BUF = UPLIT(BYTE(%ASCIC ' Old Rec:      '));
: 1149      2441 2
: 1150      2442 2 OWN
: 1151      2443 2     BANNER_VEC : VECTOR[NSASK_RECTYPNUM_NETUAF]
: 1152      2444 2     INITIAL(UPLIT(BYTE(%ASCIC ' / Network UAF record addition')),
: 1153      2445 2     UPLIT(BYTE(%ASCIC ' / Network UAF record deletion')),
: 1154      2446 2     UPLIT(BYTE(%ASCIC ' / Network UAF record modification'))),
: 1155      2447 2     LABEL_VEC : VECTOR[NSASK_RECTYPNUM_NETUAF]
: 1156      2448 2     INITIAL(UPLIT(BYTE(%ASCIC ' Rec Add:      ')),
: 1157      2449 2     UPLIT(BYTE(%ASCIC ' Rec Del:      ')),
: 1158      2450 2     UPLIT(BYTE(%ASCIC ' New Rec:      ')));
: 1159      2451 2 LOCAL
: 1160      2452 2     STATUS : LONG;
: 1161      2453 2
: 1162      2454 2 !
: 1163      2455 2 ! Format the common message lines.
: 1164      2456 2
: 1165      2457 2 STATUS = PUT_COMMON(.RECORD_PTR, .BANNER_VEC[.RECORD_PTR[NSASW_REC_SUBTYPE]-1]);
: 1166      2458 2 IF NOT .STATUS THEN RETURN .STATUS;
: 1167      2459 2
: 1168      2460 2 ! Format the first network user authorization line.
: 1169      2461 2
: 1170      2462 2 PUT_STRING_ASCII(.LABEL_VEC[.RECORD_PTR[NSASW_REC_SUBTYPE]-1]);
: 1171      2463 2 PUT_NETUAF_LINE(.RECORD_PTR, 1);
```



```
: 1172      2464 2 1
: 1173      2465 2 1 If this is a network UAF modified record, format the second network user
: 1174      2466 2 1 authorization line.
: 1175      2467 2 1
: 1176      2468 2 1 IF .RECORD_PTR[NSASW_REC_SUBTYPE] EQLU NSASK_RECTYP_NETUAF_MOD
: 1177      2469 2 1 THEN
: 1178      2470 2 1 BEGIN
: 1179      2471 2 1 PUT_STRING_ASCII(OLDREC_);
: 1180      2472 2 1 PUT_NETUAF_LINE(.RECORD_PTR, 2);
: 1181      2473 2 1 END;
: 1182      2474 2 1 RETURN $$$_NORMAL;
: 1183      2475 1 1 END;
```

! End of FORMAT_NETUAF

```
.PSECT $SPLITS,NOWRT,NOEXE,2
09 3A 63 65 52 20 64 6C 4F 09 0A 0013D P.AAS: .BLKB 3
6F 69 74 69 64 64 61 20 64 72 6F 63 65 72 1E 00140 P.AAT: .ASCII <10><9>\Old Rec:\<9>
6F 69 74 69 64 64 61 20 64 72 6F 63 65 72 20 0014B P.AAT: .BLKB 1
6F 69 74 69 64 64 61 20 64 72 6F 63 65 72 20 0014C P.AAT: .ASCII <30>\ / Network UAF record addition\
6F 69 74 69 64 64 61 20 64 72 6F 63 65 72 20 0015B
6F 69 74 69 64 64 61 20 64 72 6F 63 65 72 20 0016A
6F 69 74 69 64 64 61 20 64 72 6F 63 65 72 20 0016B
6F 69 74 69 64 64 61 20 64 72 6F 63 65 72 20 0016C P.AAU: .BLKB 1
6F 69 74 69 64 64 61 20 64 72 6F 63 65 72 20 0017B P.AAU: .ASCII <30>\ / Network UAF record deletion\
6F 69 74 69 64 64 61 20 64 72 6F 63 65 72 20 0018A
6F 69 74 69 64 64 61 20 64 72 6F 63 65 72 20 0018B
6F 69 74 69 64 64 61 20 64 72 6F 63 65 72 20 0018C P.AAV: .BLKB 1
6F 69 74 69 64 64 61 20 64 72 6F 63 65 72 20 0019B P.AAV: .ASCII \' / Network UAF record modification\
6F 69 74 69 64 64 61 20 64 72 6F 63 65 72 20 001AA
6F 69 74 69 64 64 61 20 64 72 6F 63 65 72 20 001AF
09 3A 64 64 41 20 63 65 52 09 0A 001B0 P.AAW: .BLKB 1
09 3A 6C 65 44 20 63 65 52 09 0A 001BB P.AAW: .ASCII <10><9>\Rec Add:\<9>
09 3A 6C 65 44 20 63 65 52 09 0A 001BC P.AAX: .BLKB 1
09 3A 6C 65 44 20 63 65 52 09 0A 001C7 P.AAX: .ASCII <10><9>\Rec Del:\<9>
09 3A 63 65 52 20 77 65 4E 09 0A 001C8 P.AAY: .BLKB 1
09 3A 63 65 52 20 77 65 4E 09 0A 001C8 P.AAY: .ASCII <10><9>\New Rec:\<9>
```

.PSECT \$OWNS,NOEXE,2

```
00000000' 00000000' 00000000' 00B20 BANNER_VEC:
00000000' 00000000' 00000000' 00B2C LABEL_VEC:
ADDRESS P.AAT, P.AAU, P.AAV
ADDRESS P.AAW, P.AAX, P.AAY
```

OLDREC_BUF= P.AAS

.PSECT \$CODE\$,NOWRT,2

```
0004 00000 FORMAT_NETUAF:
52 04 AC DO 00002 .WORD Save R2
50 02 A2 3C 00006 .MOVL RECORD_PTR, R2
0000' CF 40 DD 0000A .MOVZWL 2(R2), -R0
52 DD 0000F .PUSHL BANNER_VEC-4[R0]
0000V CF 02 FB 00011 .PUSHL R2
CALLS #2, PUT_COMMON
```

: 2407
: 2457
:
:
:
:

	32		50	E9	00016	BLBC	STATUS, 2\$	:	2458
	50	02	A2	3C	00019	MOVZWL	2(R2), R0	:	2462
0000V	CF	0000'	CF40	DD	0001D	PUSHL	LABEL VEC-4[R0]	:	
			01	FB	00022	CALLS	#1, PUT_STRING_ASCII	:	
			01	DD	00027	PUSHL	#1	:	2463
			52	DD	00029	PUSHL	R2	:	
0000V	CF		02	FB	0002B	CALLS	#2, PUT_NETUAF_LINE	:	
	03	02	A2	B1	00030	CMPW	2(R2), #3	:	2468
			12	12	00034	BNEQ	1\$	:	
		0000'	CF	9F	00036	PUSHAB	OLDREC BUF	:	2471
0000V	CF		01	FB	0003A	CALLS	#1, PUT_STRING_ASCII	:	
			02	DD	0003F	PUSHL	#2	:	2472
			52	DD	00041	PUSHL	R2	:	
0000V	CF		02	FB	00043	CALLS	#2, PUT_NETUAF_LINE	:	
	50		01	D0	00048	MOVL	#1, R0	:	2474
			04	0004B	2\$:	RET		:	2475

; Routine Size: 76 bytes, Routine Base: \$CODE\$ + 0492


```

1185 2476 1 ROUTINE FORMAT_LOGB(RECORD_PTR) =
1186 2477 1
1187 2478 1 ++
1188 2479 1 FUNCTIONAL DESCRIPTION:
1189 2480 1
1190 2481 1 This routine formats security alarm messages for records whose type
1191 2482 1 is NSASK_RECTYP_LOGB. (Login breakin detection)
1192 2483 1
1193 2484 1 INPUT:
1194 2485 1 RECORD_PTR : The security alarm record address.
1195 2486 1
1196 2487 1 IMPLICIT INPUT:
1197 2488 1 NONE.
1198 2489 1
1199 2490 1 OUTPUT:
1200 2491 1 NONE.
1201 2492 1
1202 2493 1 IMPLICIT OUTPUT:
1203 2494 1 Formatted message in the alarm buffer.
1204 2495 1
1205 2496 1 SIDE EFFECTS:
1206 2497 1 NONE.
1207 2498 1
1208 2499 1 ROUTINE VALUE:
1209 2500 1 SSS_NORMAL or a $FAO system service error.
1210 2501 1 --
1211 2502 1
1212 2503 2 BEGIN ! Start of FORMAT_LOGB
1213 2504 2
1214 2505 2 MAP
1215 2506 2 RECORD_PTR : REF $BBLOCK;
1216 2507 2
1217 2508 2 OWN
1218 2509 2 BANNER_VEC : VECTOR[NSASK_RECTYPNUM_LOGB]
1219 2510 2 INITIAL(UPLIT(BYTE(%ASCIC ' / Dialup interactive breakin detection')),
1220 2511 2 UPLIT(BYTE(%ASCIC ' / Local interactive breakin detection')),
1221 2512 2 UPLIT(BYTE(%ASCIC ' / Remote interactive breakin detection')),
1222 2513 2 UPLIT(BYTE(%ASCIC ' / Network breakin detection')),
1223 2514 2 UPLIT(BYTE(%ASCIC ' / Detached process breakin detection')));
1224 2515 2
1225 2516 2 LOCAL
1226 2517 2 STATUS : LONG;
1227 2518 2
1228 2519 2
1229 2520 2 Format the common message lines.
1230 2521 2
1231 2522 2 STATUS = PUT_COMMON(.RECORD_PTR, .BANNER_VEC[.RECORD_PTR[NSASK_REC_SUBTYPE]-1]);
1232 2523 2 IF NOT .STATUS THEN RETURN .STATUS;
1233 2524 2 STATUS = PUT_PKT_PASSWORD(.RECORD_PTR);
1234 2525 2 IF NOT .STATUS THEN RETURN .STATUS;
1235 2526 2
1236 2527 2 Dispatch by record subtype to format the remainder of the message lines.
1237 2528 2
1238 2529 2 CASE .RECORD_PTR[NSASK_REC_SUBTYPE] FROM 1 TO NSASK_RECTYPNUM_LOGB OF
1239 2530 2 SET
1240 2531 2 [NSASK_RECTYP_LOGB_DIA, NSASK_RECTYP_LOGB_LOC] :
1241 2532 2

```



```
: 1242      2533      2      ! Dialup and local interactive breakin detection.
: 1243      2534      2
: 1244      2535      2      BEGIN
: 1245      2536      2      PUT_DEVNAM_LINE(.RECORD_PTR);
: 1246      2537      2      END;
: 1247      2538      2      [NSASK_RECTYP_LOGB_REM] :
: 1248      2539      2
: 1249      2540      2      ! Remote interactive breakin detection.
: 1250      2541      2
: 1251      2542      2      BEGIN
: 1252      2543      2      PUT_DEVNAM_LINE(.RECORD_PTR);
: 1253      2544      2      STATUS = PUT_SOURCE_LINE(.RECORD_PTR);
: 1254      2545      2      IF NOT .STATUS THEN RETURN .STATUS;
: 1255      2546      2      END;
: 1256      2547      2      [NSASK_RECTYP_LOGB_NET] :
: 1257      2548      2
: 1258      2549      2      ! Network breakin detection.
: 1259      2550      2
: 1260      2551      2      BEGIN
: 1261      2552      2      STATUS = PUT_SOURCE_LINE(.RECORD_PTR);
: 1262      2553      2      IF NOT .STATUS THEN RETURN .STATUS;
: 1263      2554      2      END;
: 1264      2555      2      [NSASK_RECTYP_LOGB_DET] :
: 1265      2556      2
: 1266      2557      2      ! Detached process breakin detection.
: 1267      2558      2
: 1268      2559      2      BEGIN
: 1269      2560      2      PUT_PKT_PARENTUN(.RECORD_PTR);
: 1270      2561      2      END;
: 1271      2562      2      TES;
: 1272      2563      2      RETURN $$$_NORMAL;
: 1273      2564      1      END;

! End of FORMAT_LOGB
```

```
.PSECT $SPLIT$,NOWRT,NOEXE,2

65 74 6E 69 20 70 75 6C 61 69 44 20 2F 20 27 001D3
6E 69 6B 61 65 72 62 20 65 76 69 74 63 61 72 001D4 P.AAZ: .BLKB 1
: 72 65 74 6E 69 20 6C 61 63 6F 4C 20 2F 20 26 001E3 :
: 20 6E 69 6B 61 65 72 62 20 65 76 69 74 63 61 001F2 :
: 6E 6F 69 74 63 65 74 65 64 20 001FC P.ABA: .ASCII \% / Local interactive breakin detection\ :
: 0020B :
: 0021A :
: 00223 :
65 74 6E 69 20 65 74 6F 6D 65 52 20 2F 20 27 00224 P.ABB: .BLKB 1
6E 69 6B 61 65 72 62 20 65 76 69 74 63 61 72 00233 :
: 6E 6F 69 74 63 65 74 65 64 20 00242 :
65 72 62 20 6B 72 6F 77 74 65 4E 20 2F 20 1C 0024C P.ABC: .ASCII <28>\ / Network breakin detection\ :
: 6E 6F 69 74 63 65 74 65 64 20 6E 69 6B 61 0025B :
: 00269 :
72 70 20 64 65 6B 63 61 74 65 44 20 2F 20 25 0026C P.ABD: .BLKB 3
64 20 6E 69 6B 61 65 72 62 20 73 73 65 63 6F 0027B :
: 6E 6F 69 74 63 65 74 65 0028A :
:

.PSECT $OWNS$,NOEXE,2
```


00000000' 00000000' 00000000' 00000000' 00000000' 00B38 BANNER_VEC:
.ADDRESS P.AAZ, P.ABA, P.ABB, P.ABC, P.ABD ;

.PSECT \$CODE\$,NOWRT,2

		000C 00000		FORMAT_LOGB:		
	52	04	AC	D0 00002	.WORD	Save R2,R3 : 2476
	50	02	A2	3C 00006	MOVL	RECORD_PTR, R2 : 2522
		0000'	CF	40 DD 0000A	MOVZWL	2(R2), R0
				52 DD 0000F	PUSHL	BANNER_VEC-4[R0]
0000V	CF		02	FB 00011	PUSHL	R2
	53		50	D0 00016	CALLS	#2, PUT_COMMON
	39		53	E9 00019	MOVL	R0, STATUS
			52	DD 0001C	BLBC	STATUS, 5\$: 2523
0000V	CF		01	FB 0001E	PUSHL	R2 : 2524
	53		50	D0 00023	CALLS	#1, PUT_PKT_PASSWORD
	2C		53	E9 00026	MOVL	R0, STATUS
	01		02	AF 00029	BLBC	STATUS, 5\$: 2525
001A	0013	000A	000A	0002E 1\$:	CASEW	2(R2), #1, #4 : 2529
			002B	00036	.WORD	2\$-1\$,-
						2\$-1\$,-
						3\$-1\$,-
						4\$-1\$,-
						6\$-1\$
			52	DD 00038 2\$:	PUSHL	R2 : 2536
0000V	CF		01	FB 0003A	CALLS	#1, PUT_DEVNAM_LINE
			1F	11 0003F	BRB	7\$: 2529
			52	DD 00041 3\$:	PUSHL	R2 : 2543
0000V	CF		01	FB 00043	CALLS	#1, PUT_DEVNAM_LINE
			52	DD 00048 4\$:	PUSHL	R2 : 2552
0000V	CF		01	FB 0004A	CALLS	#1, PUT_SOURCE_LINE
	53		50	D0 0004F	MOVL	R0, STATUS
	0B		53	E8 00052	BLBS	STATUS, 7\$: 2553
	50		53	D0 00055 5\$:	MOVL	STATUS, R0
				04 00058	RET	
			52	DD 00059 6\$:	PUSHL	R2 : 2560
0000V	CF		01	FB 0005B	CALLS	#1, PUT_PKT_PARENTUN
	50		01	D0 00060 7\$:	MOVL	#1, R0 : 2563
			04	00063	RET	: 2564

; Routine Size: 100 bytes, Routine Base: \$CODE\$ + 04DE


```
: 1275 2565 1 ROUTINE FORMAT_LOGI(RECORD_PTR) =
: 1276 2566 1
: 1277 2567 1 !++
: 1278 2568 1 FUNCTIONAL DESCRIPTION:
: 1279 2569 1
: 1280 2570 1 This routine formats security alarm messages for records whose type
: 1281 2571 1 is NSASK_RECTYP_LOGI. (Successful logins)
: 1282 2572 1
: 1283 2573 1 INPUT:
: 1284 2574 1 RECORD_PTR : The security alarm record address.
: 1285 2575 1
: 1286 2576 1 IMPLICIT INPUT:
: 1287 2577 1 NONE.
: 1288 2578 1
: 1289 2579 1 OUTPUT:
: 1290 2580 1 NONE.
: 1291 2581 1
: 1292 2582 1 IMPLICIT OUTPUT:
: 1293 2583 1 Formatted message in the alarm buffer.
: 1294 2584 1
: 1295 2585 1 SIDE EFFECTS:
: 1296 2586 1 NONE.
: 1297 2587 1
: 1298 2588 1 ROUTINE VALUE:
: 1299 2589 1 $$$_NORMAL or a $FAO system service error.
: 1300 2590 1 --
: 1301 2591 1
: 1302 2592 2 BEGIN ! Start of FORMAT_LOGI
: 1303 2593 2
: 1304 2594 2 MAP
: 1305 2595 2 RECORD_PTR : REF $BBLOCK;
: 1306 2596 2
: 1307 2597 2 OWN
: 1308 2598 2 BANNER_VEC : VECTOR[NSASK_RECTYPNUM_LOGI]
: 1309 2599 2 INITIAL(UPLIT(BYTE(%ASCIC ' / Batch process login')),
: 1310 2600 2 UPLIT(BYTE(%ASCIC ' / Dialup interactive login')),
: 1311 2601 2 UPLIT(BYTE(%ASCIC ' / Local interactive login')),
: 1312 2602 2 UPLIT(BYTE(%ASCIC ' / Remote interactive login')),
: 1313 2603 2 UPLIT(BYTE(%ASCIC ' / Network login')),
: 1314 2604 2 UPLIT(BYTE(%ASCIC ' / Subprocess login')),
: 1315 2605 2 UPLIT(BYTE(%ASCIC ' / Detached process login')));
: 1316 2606 2
: 1317 2607 2 LOCAL
: 1318 2608 2 STATUS : LONG;
: 1319 2609 2
: 1320 2610 2 !
: 1321 2611 2 Format the common message lines.
: 1322 2612 2
: 1323 2613 2 STATUS = PUT_COMMON(.RECORD_PTR, .BANNER_VEC[.RECORD_PTR[NSA$W_REC_SUBTYPE]-1]);
: 1324 2614 2 IF NOT .STATUS THEN RETURN .STATUS;
: 1325 2615 2
: 1326 2616 2 Dispatch by record subtype to format the remainder of the message lines.
: 1327 2617 2
: 1328 2618 2 CASE .RECORD_PTR[NSA$W_REC_SUBTYPE] FROM 1 TO NSASK_RECTYPNUM_LOGI OF
: 1329 2619 2 SET
: 1330 2620 2 [NSASK_RECTYP_LOGI_BAT] :
: 1331 2621 2 !
```



```
.PSECT SPLITS,NOWRT,NOEXE,2
```

65	63	6F	72	70	20	68	63	74	61	42	20	2F	20	16	00292		.BLKB	2
							6E	69	67	6F	6C	20	73	73	00294	P.ABE:	.ASCII	<22>\ / Batch process login\
															002A3			
															002AB		.BLKB	1
65	74	6E	69	20	70	75	6C	61	69	44	20	2F	20	1B	002AC	P.ABF:	.ASCII	<27>\ / Dialup interactive login\


```
72 65 74 6E 69 67 6F 6C 20 65 76 69 74 63 61 72 002BB P.ABG: .ASCII <26>\ / Local interactive login\
72 65 74 6E 69 67 6F 6C 20 65 76 69 74 63 61 72 002C8 P.ABH: .BLKB 1
65 74 6E 69 20 65 74 6F 6D 65 52 20 2F 20 1B 002E4 P.ABI: .ASCII <27>\ / Remote interactive login\
67 6F 6C 20 6B 72 6F 77 74 65 4E 20 2F 20 10 002F3 P.ABJ: .ASCII <16>\ / Network login\
20 73 73 65 63 6F 72 70 62 75 53 20 2F 20 13 00300 P.ABK: .BLKB 3
72 70 20 64 65 68 63 61 74 65 44 20 2F 20 19 00311 P.ABJ: .ASCII <19>\ / Subprocess login\
6E 69 67 6F 6C 20 73 73 65 63 6F 00314 P.ABK: .ASCII <25>\ / Detached process login\
00323
00328
00337
```

.PSECT \$OWN\$,NOEXE,2

```
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00B4C BANNER_VEC:
00000000' 00B64 .ADDRESS P.ABE, P.ABF, P.ABG, P.ABH, P.ABI, -
P.ABJ, P.ABK
```

.PSECT \$CODE\$,NOWRT,2

```
000C 00000 FORMAT_LOGI:
52 04 AC D0 00002 .WORD Save R2,R3
50 02 A2 3C 00006 MOVL RECORD_PTR, R2
0000' CF 40 DD 0000A MOVZWL 2(R2), R0
52 DD 0000F PUSHL BANNER_VEC-4[R0]
0000V CF 02 FB 00011 PUSHL R2
53 50 D0 00016 CALLS #2, PUT_COMMON
3B 53 E9 00019 MOVL R0, STATUS
0019 06 01 02 A2 AF 0001C BLBC STATUS, 7$
0010 0010 0041 00021 CASEW 2(R2), #1, #6
003A 0029 0020 00029 .WORD 9$-1$,-
2$-1$,-
2$-1$,-
3$-1$,-
4$-1$,-
5$-1$,-
8$-1$
9$
0000V CF 31 11 0002F BRB R2
52 DD 00031 2$: PUSHL R2
01 FB 00033 CALLS #1, PUT_DEVNAM_LINE
28 11 00038 BRB 9$
52 DD 0003A 3$: PUSHL R2
0000V CF 01 FB 0003C CALLS #1, PUT_DEVNAM_LINE
52 DD 00041 4$: PUSHL R2
0000V CF 01 FB 00043 CALLS #1, PUT_SOURCE_LINE
07 11 00048 BRB 6$
52 DD 0004A 5$: PUSHL R2
0000V CF 01 FB 0004C CALLS #1, PUT_PKT_PARENTPID
53 50 D0 00051 6$: MOVL R0, STATUS
53 E8 00054 BLBS STATUS, 9$
53 D0 00057 7$: MOVL STATUS, R0
04 0005A RET
52 DD 0005B 8$: PUSHL R2
```


OPCSSECURITY
V04-000

J 16
16-Sep-1984 01:48:50
14-Sep-1984 12:50:55

VAX-11 Bliss-32 V4.0-742
[OPCOM.SRC]SECURITY.B32;1

Page 45
(10)

0000V CF
50

01 FB 0005D
01 D0 00062 9\$:
04 00065

CALLS #1, PUT_PKT_PARENTUN
MOVL #1, R0
RET

:
: 2667
: 2668

; Routine Size: 102 bytes, Routine Base: \$CODE\$ + 0542


```
1380 2669 1 ROUTINE FORMAT_LOGF(RECORD_PTR) =
1381 2670 1
1382 2671 1 ++
1383 2672 1 FUNCTIONAL DESCRIPTION:
1384 2673 1
1385 2674 1 This routine formats security alarm messages for records whose type
1386 2675 1 is NSASK_RECTYP_LOGF. (Login failures)
1387 2676 1
1388 2677 1 INPUT:
1389 2678 1 RECORD_PTR : The security alarm record address.
1390 2679 1
1391 2680 1 IMPLICIT INPUT:
1392 2681 1 NONE.
1393 2682 1
1394 2683 1 OUTPUT:
1395 2684 1 NONE.
1396 2685 1
1397 2686 1 IMPLICIT OUTPUT:
1398 2687 1 Formatted message in the alarm buffer.
1399 2688 1
1400 2689 1 SIDE EFFECTS:
1401 2690 1 NONE.
1402 2691 1
1403 2692 1 ROUTINE VALUE:
1404 2693 1 $$$_NORMAL or a $FAO system service error.
1405 2694 1 --
1406 2695 1
1407 2696 2 BEGIN ! Start of FORMAT_LOGF
1408 2697 2
1409 2698 2 MAP
1410 2699 2 RECORD_PTR : REF $BLOCK;
1411 2700 2
1412 2701 2 OWN
1413 2702 2 BANNER_VEC : VECTOR[NSASK_RECTYPNUM_LOGF]
1414 2703 2 INITIAL(UPLIT(BYTE(%ASCIC ' / Batch process login failure')),
1415 2704 2 UPLIT(BYTE(%ASCIC ' / Dialup interactive login failure')),
1416 2705 2 UPLIT(BYTE(%ASCIC ' / Local interactive login failure')),
1417 2706 2 UPLIT(BYTE(%ASCIC ' / Remote interactive login failure')),
1418 2707 2 UPLIT(BYTE(%ASCIC ' / Network login failure')),
1419 2708 2 UPLIT(BYTE(%ASCIC ' / Subprocess login failure')),
1420 2709 2 UPLIT(BYTE(%ASCIC ' / Detached process login failure')));
1421 2710 2
1422 2711 2 LOCAL
1423 2712 2 STATUS : LONG;
1424 2713 2
1425 2714 2 !
1426 2715 2 Format the common message lines.
1427 2716 2
1428 2717 2 STATUS = PUT COMMON(.RECORD_PTR, .BANNER_VEC[.RECORD_PTR[NSASW_REC_SUBTYPE]-1]);
1429 2718 2 IF NOT .STATUS THEN RETURN .STATUS;
1430 2719 2 STATUS = PUT PKT STATUS(.RECORD_PTR);
1431 2720 2 IF NOT .STATUS THEN RETURN .STATUS;
1432 2721 2
1433 2722 2 Dispatch by record subtype to format the remainder of the message lines.
1434 2723 2
1435 2724 2 CASE .RECORD_PTR[NSASW_REC_SUBTYPE] FROM 1 TO NSASK_RECTYPNUM_LOGF OF
1436 2725 2 SET
```



```
! End of FORMAT_LOGF
```

```

65 63 6F 72 70 20 68 63 74 61 42 20 2F 20 1E 00342
72 75 6C 69 61 66 20 6E 69 67 6F 6C 20 73 73 00344 P.ABL: .BLKB 2
                                .ASCII <30>\ / Batch process login failure\
                                00353

```



```
65 74 6E 69 20 70 75 6C 61 69 44 20 2F 20 23 00362
66 20 6E 69 67 6F 6C 20 65 76 69 74 63 61 72 00363
72 65 74 6E 69 20 6C 61 63 6F 4C 20 2F 20 22 00364 P.ABM: .BLKB 1
61 66 20 6E 69 67 6F 6C 20 65 76 69 74 63 61 72 00373 P.ABM: .ASCII \# / Dialup interactive login failure\
72 65 74 6E 69 20 6C 61 63 6F 4C 20 2F 20 22 00382 P.ABN: .BLKB 1
61 66 20 6E 69 67 6F 6C 20 65 76 69 74 63 61 72 00388 P.ABN: .ASCII \" / Local interactive login failure\
65 74 6E 69 20 65 74 6F 6D 65 52 20 2F 20 23 00397 P.ABN: .ASCII \" / Local interactive login failure\
66 20 6E 69 67 6F 6C 20 65 76 69 74 63 61 72 003A6 P.ABO: .BLKB 1
67 6F 6C 20 6B 72 6F 77 74 65 52 20 2F 20 23 003AB P.ABO: .ASCII \# / Remote interactive login failure\
65 74 6E 69 20 65 74 6F 6D 65 52 20 2F 20 23 003AC P.ABO: .ASCII \# / Remote interactive login failure\
66 20 6E 69 67 6F 6C 20 65 76 69 74 63 61 72 003BB P.ABO: .ASCII \# / Remote interactive login failure\
67 6F 6C 20 6B 72 6F 77 74 65 52 20 2F 20 23 003CA P.ABP: .BLKB 3
65 74 6E 69 20 65 74 6F 6D 65 52 20 2F 20 23 003D0 P.ABP: .ASCII <24>\ / Network login failure\
66 20 6E 69 67 6F 6C 20 65 76 69 74 63 61 72 003DF P.ABP: .ASCII <24>\ / Network login failure\
67 6F 6C 20 6B 72 6F 77 74 65 52 20 2F 20 23 003E9 P.ABQ: .BLKB 3
20 73 73 65 63 6F 72 70 62 75 53 20 2F 20 1B 003EC P.ABQ: .ASCII <27>\ / Subprocess login failure\
72 70 20 64 65 68 63 61 74 65 44 20 2F 20 21 003FB P.ABQ: .ASCII <27>\ / Subprocess login failure\
69 61 66 20 6E 69 67 6F 6C 20 73 73 65 63 6F 6C 00408 P.ABR: .ASCII \! / Detached process login failure\
65 74 6E 69 20 65 74 6F 6D 65 52 20 2F 20 23 00417 P.ABR: .ASCII \! / Detached process login failure\
66 20 6E 69 67 6F 6C 20 65 76 69 74 63 61 72 00426 P.ABR: .ASCII \! / Detached process login failure\
67 6F 6C 20 6B 72 6F 77 74 65 52 20 2F 20 23 00426 P.ABR: .ASCII \! / Detached process login failure\
```

.PSECT \$OWNS\$,NOEXE,2

```
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00B68 BANNER_VEC:
00000000' 00B80 .ADDRESS P.ABL, P.ABM, P.ABN, P.ABO, P.ABP, -
P.ABQ, P.ABR
```

.PSECT \$CODE\$,NOWRT,2

```
000C 00000 FORMAT_LOGF:
52 04 AC D0 00002 .WORD Save R2,R3
50 02 A2 3C 00006 MOVL RECORD_PTR, R2
0000' CF 40 DD 0000A MOVZWL 2(R2), R0
52 DD 0000F PUSHBL BANNER_VEC-4[R0]
0000V CF 02 FB 00011 PUSHBL R2
53 50 D0 00016 CALLS #2, PUT_COMMON
48 53 E9 00019 MOVL R0, STATUS
0000V CF 52 DD 0001C BLBC STATUS, 7$
53 50 D0 00023 PUSHBL R2
3B 53 E9 00026 CALLS #1, PUT_PKT_STATUS
01 02 A2 AF 00029 MOVL R0, STATUS
0019 0010 0041 0002E BLBC STATUS, 7$
003A 0029 0020 00036 CASEW 2(R2), #1, #6
1$: .WORD 9$-1$, -
2$-1$, -
2$-1$, -
3$-1$, -
4$-1$, -
5$-1$, -
8$-1$
9$
0000V CF 31 11 0003C BRB 9$
52 DD 0003E 2$: PUSHBL R2
01 FB 00040 CALLS #1, PUT_DEVNAM_LINE
28 11 00045 BRB 9$
52 DD 00047 3$: PUSHBL R2
```


OPC\$SECURITY
V04-000

B 1
16-Sep-1984 01:48:50 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:50:55 [OPCOM.SRC]SECURITY.B32;1

Page 49
(11)

0000V	CF	01	FB	00049	CALLS	#1, PUT_DEVNAM_LINE	:	
		52	DD	0004E	PUSHL	R2	:	2754
0000V	CF	01	FB	00050	CALLS	#1, PUT_SOURCE_LINE	:	
		07	11	00055	BRB	6\$	:	
		52	DD	00057	PUSHL	R2	:	2762
0000V	CF	01	FB	00059	CALLS	#1, PUT_PKT_PARENTPID	:	
	53	50	DO	0005E	MOVL	R0, STATUS	:	
	OB	53	E8	00061	BLBS	STATUS, 9\$	:	2763
	50	53	DO	00064	MOVL	STATUS, R0	:	
			04	00067	RET		:	
		52	DD	00068	PUSHL	R2	:	2770
0000V	CF	01	FB	0006A	CALLS	#1, PUT_PKT_PARENTUN	:	
	50	01	DO	0006F	MOVL	#1, R0	:	2773
			04	00072	RET		:	2774

; Routine Size: 115 bytes, Routine Base: \$CODE\$ + 05A8


```

: 1487 2775 1 ROUTINE FORMAT_LOGO(RECORD_PTR) =
: 1488 2776 1
: 1489 2777 1 !++
: 1490 2778 1 FUNCTIONAL DESCRIPTION:
: 1491 2779 1
: 1492 2780 1 This routine formats security alarm messages for records whose type
: 1493 2781 1 is NSASK_RECTYP_LOGO. (Logouts)
: 1494 2782 1
: 1495 2783 1 INPUT:
: 1496 2784 1 RECORD_PTR : The security alarm record address.
: 1497 2785 1
: 1498 2786 1 IMPLICIT INPUT:
: 1499 2787 1 NONE.
: 1500 2788 1
: 1501 2789 1 OUTPUT:
: 1502 2790 1 NONE.
: 1503 2791 1
: 1504 2792 1 IMPLICIT OUTPUT:
: 1505 2793 1 Formatted message in the alarm buffer.
: 1506 2794 1
: 1507 2795 1 SIDE EFFECTS:
: 1508 2796 1 NONE.
: 1509 2797 1
: 1510 2798 1 ROUTINE VALUE:
: 1511 2799 1 $$$_NORMAL or a $FAO system service error.
: 1512 2800 1 --
: 1513 2801 1
: 1514 2802 2 BEGIN ! Start of FORMAT_LOGO
: 1515 2803 2
: 1516 2804 2 MAP
: 1517 2805 2 RECORD_PTR : REF $BBLOCK;
: 1518 2806 2
: 1519 2807 2 OWN
: 1520 2808 2 BANNER_VEC : VECTOR[NSASK_RECTYPNUM_LOGO]
: 1521 2809 2 INITIAL(UPLIT(BYTE(%ASCIC ' / Batch process logout')),
: 1522 2810 2 UPLIT(BYTE(%ASCIC ' / Dialup interactive logout')),
: 1523 2811 2 UPLIT(BYTE(%ASCIC ' / Local interactive logout')),
: 1524 2812 2 UPLIT(BYTE(%ASCIC ' / Remote interactive logout')),
: 1525 2813 2 UPLIT(BYTE(%ASCIC ' / Network logout')),
: 1526 2814 2 UPLIT(BYTE(%ASCIC ' / Subprocess logout')),
: 1527 2815 2 UPLIT(BYTE(%ASCIC ' / Detached process logout')));
: 1528 2816 2
: 1529 2817 2 LOCAL
: 1530 2818 2 STATUS : LONG;
: 1531 2819 2
: 1532 2820 2
: 1533 2821 2 Format the common message lines.
: 1534 2822 2
: 1535 2823 2 STATUS = PUT_COMMON(.RECORD_PTR, .BANNER_VEC[.RECORD_PTR[NSA$W_REC_SUBTYPE]-1]);
: 1536 2824 2 IF NOT .STATUS THEN RETURN .STATUS;
: 1537 2825 2
: 1538 2826 2 Dispatch by record subtype to format the remainder of the message lines.
: 1539 2827 2
: 1540 2828 2 CASE .RECORD_PTR[NSA$W_REC_SUBTYPE] FROM 1 TO NSASK_RECTYPNUM_LOGO OF
: 1541 2829 2 SET
: 1542 2830 2 [NSASK_RECTYP_LOGO_BAT] :
: 1543 2831 2

```



```
: 1544      2832      2      | Batch process logout.
: 1545      2833      2
: 1546      2834      3      | BEGIN
: 1547      2835      3      | SSS_NORMAL;
: 1548      2836      2      | END;
: 1549      2837      2      | [NSASK_RECTYP_LOGO_DIA, NSASK_RECTYP_LOGO_LOC] :
: 1550      2838      2
: 1551      2839      2      | Dialup and local interactive logout.
: 1552      2840      2
: 1553      2841      3      | BEGIN
: 1554      2842      3      | PUT_DEVNAM_LINE(.RECORD_PTR);
: 1555      2843      2      | END;
: 1556      2844      2      | [NSASK_RECTYP_LOGO_REM] :
: 1557      2845      2
: 1558      2846      2      | Remote interactive logout.
: 1559      2847      2
: 1560      2848      3      | BEGIN
: 1561      2849      3      | PUT_DEVNAM_LINE(.RECORD_PTR);
: 1562      2850      3      | STATUS = PUT_SOURCE_LINE(.RECORD_PTR);
: 1563      2851      3      | IF NOT .STATUS THEN RETURN .STATUS;
: 1564      2852      2      | END;
: 1565      2853      2      | [NSASK_RECTYP_LOGO_NET] :
: 1566      2854      2
: 1567      2855      2      | Network logout.
: 1568      2856      2
: 1569      2857      3      | BEGIN
: 1570      2858      3      | STATUS = PUT_SOURCE_LINE(.RECORD_PTR);
: 1571      2859      3      | IF NOT .STATUS THEN RETURN .STATUS;
: 1572      2860      2      | END;
: 1573      2861      2      | [NSASK_RECTYP_LOGO_SUB] :
: 1574      2862      2
: 1575      2863      2      | Subprocess logout.
: 1576      2864      2
: 1577      2865      3      | BEGIN
: 1578      2866      3      | STATUS = PUT_PKT_PARENTPID(.RECORD_PTR);
: 1579      2867      3      | IF NOT .STATUS THEN RETURN .STATUS;
: 1580      2868      2      | END;
: 1581      2869      2      | [NSASK_RECTYP_LOGO_DET] :
: 1582      2870      2
: 1583      2871      2      | Detached process logout.
: 1584      2872      2
: 1585      2873      3      | BEGIN
: 1586      2874      3      | SSS_NORMAL;
: 1587      2875      2      | END;
: 1588      2876      2      | TES;
: 1589      2877      2      | RETURN SSS_NORMAL;
: 1590      2878      1      | END;

! End of FORMAT_LOGO
```

.PSECT \$SPLITS,NOWRT,NOEXE,2

65	63	6F	72	70	20	68	63	74	61	42	20	2F	20	17	0042A				
						74	75	6F	67	6F	6C	20	73	73	0042C	P.ABS:	.BLKB	2	
															0043B		.ASCII	<23>\	/ Batch process logout\
65	74	6E	69	20	70	75	6C	61	69	44	20	2F	20	1C	00444	P.ABT:	.ASCII	<28>\	/ Dialup interactive logout\
	74	75	6F	67	6F	6C	20	65	76	69	74	63	61	72	00453				


```

72 65 74 6E 69 20 6C 61 63 6F 4C 20 2F 20 1B 00461
65 74 74 75 6F 67 6F 6C 20 65 76 69 74 63 61 00464 P.ABU: .BLKB 3
74 75 6F 67 6F 6C 20 65 76 69 74 63 61 72 00473 P.ABV: .ASCII <27>\ / Local interactive logout\
67 6F 6C 20 6B 72 6F 77 74 65 4E 20 2F 20 11 00480 P.ABW: .ASCII <28>\ / Remote interactive logout\
74 75 6F 67 6F 6C 20 65 76 69 74 63 61 72 0048F
20 73 73 65 63 6F 72 70 62 75 53 20 2F 20 14 0049D P.ABX: .BLKB 3
74 75 6F 67 6F 6C 20 65 76 69 74 63 61 72 004A0 P.ABX: .ASCII <17>\ / Network logout\
72 70 20 64 65 68 63 61 74 65 44 20 2F 20 1A 004AF P.ABX: .BLKB 2
74 75 6F 67 6F 6C 20 65 76 69 74 63 61 72 004B2 P.ABX: .ASCII <20>\ / Subprocess logout\
004B4 P.ABX: .ASCII <20>\ / Subprocess logout\
004C3 P.ABY: .BLKB 3
004C9 P.ABY: .ASCII <26>\ / Detached process logout\
004CC P.ABY: .ASCII <26>\ / Detached process logout\
004DB

```

.PSECT \$OWN\$,NOEXE,2

```

00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00B84 BANNER_VEC:
00000000' 00B9C .ADDRESS P.ABS, P.ABT, P.ABU, P.ABV, P.ABW, -
P.ABX, P.ABY

```

.PSECT \$CODE\$,NOWRT,2

000C 00000 FORMAT_LOGO:

```

0019 06 0010 003A 0000V CF 53 04 AC D0 00002 .WORD Save R2,R3
0000V CF 50 02 A3 3C 00006 MOVL RECORD_PTR, R3
0000V CF 52 0000' CF 40 DD 0000A MOVZWL 2(R3), R0
0000V CF 53 DD 0000F PUSHL BANNER_VEC-4[R0]
0000V CF 52 02 FB 00011 PUSHL R3
0000V CF 52 50 D0 00016 CALLS #2, PUT_COMMON
0000V CF 52 52 E9 00019 MOVL R0, STATUS
0000V CF 52 02 A3 AF 0001C BLBC STATUS, 7$
0000V CF 52 003A 00021 1$ CASEW 2(R3), #1, #6
0000V CF 52 0020 00029 .WORD 8$-1$,-
2$-1$,-
2$-1$,-
3$-1$,-
4$-1$,-
5$-1$,-
8$-1$
0000V CF 53 2A 11 0002F BRB 8$
0000V CF 53 DD 00031 2$ PUSHL R3
0000V CF 53 01 FB 00033 CALLS #1, PUT_DEVNAM_LINE
0000V CF 53 21 11 00038 BRB 8$
0000V CF 53 DD 0003A 3$ PUSHL R3
0000V CF 53 01 FB 0003C CALLS #1, PUT_DEVNAM_LINE
0000V CF 53 DD 00041 4$ PUSHL R3
0000V CF 53 01 FB 00043 CALLS #1, PUT_SOURCE_LINE
0000V CF 53 07 11 00048 BRB 6$
0000V CF 53 DD 0004A 5$ PUSHL R3
0000V CF 53 01 FB 0004C CALLS #1, PUT_PKT_PARENTPID
0000V CF 52 50 D0 00051 6$ MOVL R0, STATUS
0000V CF 52 52 E8 00054 BLBS STATUS, 8$
0000V CF 52 52 D0 00057 7$ MOVL STATUS, R0
0000V CF 52 04 0005A RET

```


OPC\$SECURITY
V04-000

F 1
16-Sep-1984 01:48:50
14-Sep-1984 12:50:55

VAX-11 Bliss-32 V4.0-742
[OPCOM.SRC]SECURITY.B32;1

Page 53
(12)

50

01 D0 0005B 8\$:
04 0005E

MOVL #1, R0
RET

: 2877
: 2878

; Routine Size: 95 bytes, Routine Base: \$CODE\$ + 061B


```
: 1592      2879 1 ROUTINE FORMAT_VOL(RECORD_PTR) =
: 1593      2880 1
: 1594      2881 1 !++
: 1595      2882 1 FUNCTIONAL DESCRIPTION:
: 1596      2883 1
: 1597      2884 1 This routine formats security alarm messages for records whose type
: 1598      2885 1 is NSASK_RECTYP_VOL. (Volume operations)
: 1599      2886 1
: 1600      2887 1 INPUT:
: 1601      2888 1 RECORD_PTR : The security alarm record address.
: 1602      2889 1
: 1603      2890 1 IMPLICIT INPUT:
: 1604      2891 1 NONE.
: 1605      2892 1
: 1606      2893 1 OUTPUT:
: 1607      2894 1 NONE.
: 1608      2895 1
: 1609      2896 1 IMPLICIT OUTPUT:
: 1610      2897 1 Formatted message in the alarm buffer.
: 1611      2898 1
: 1612      2899 1 SIDE EFFECTS:
: 1613      2900 1 NONE.
: 1614      2901 1
: 1615      2902 1 ROUTINE VALUE:
: 1616      2903 1 $$$_NORMAL or a $FAO system service error.
: 1617      2904 1 --
: 1618      2905 1
: 1619      2906 2 BEGIN ! Start of FORMAT_VOL
: 1620      2907 2
: 1621      2908 2 MAP
: 1622      2909 2 RECORD_PTR : REF $BBLOCK;
: 1623      2910 2
: 1624      2911 2 OWN
: 1625      2912 2 BANNER_VEC : VECTOR[NSASK_RECTYPNUM_VOL]
: 1626      2913 2 INITIAL(UPLIT(BYTE(%ASCIC ' / Volume mount')),
: 1627      2914 2 UPLIT(BYTE(%ASCIC ' / Volume dismount')));
: 1628      2915 2
: 1629      2916 2 LOCAL
: 1630      2917 2 STATUS : LONG;
: 1631      2918 2
: 1632      2919 2 !
: 1633      2920 2 Format the common message lines.
: 1634      2921 2
: 1635      2922 2 STATUS = PUT_COMMON(.RECORD_PTR, .BANNER_VEC[.RECORD_PTR[NSA$W_REC_SUBTYPE]-1]);
: 1636      2923 2 IF NOT .STATUS THEN RETURN .STATUS;
: 1637      2924 2
: 1638      2925 2 Format the image name, device name, logical name, volume name, and volume set
: 1639      2926 2 name message lines.
: 1640      2927 2
: 1641      2928 2 PUT_PKT_IMGNAME(.RECORD_PTR);
: 1642      2929 2 PUT_PKT_DEVNAME(.RECORD_PTR);
: 1643      2930 2 PUT_PKT_LOGNAME(.RECORD_PTR);
: 1644      2931 2 PUT_PKT_VOLNAME(.RECORD_PTR);
: 1645      2932 2 IF .PKT_COUNT_TAB[NSASK_PKT_TYP_VOLSNAM-1] NEQ 0
: 1646      2933 2 THEN
: 1647      2934 2 PUT_PKT_VOLSNAM(.RECORD_PTR);
: 1648      2935 2 IF .RECORD_PTR[NSA$W_REC_SUBTYPE] EQLU NSASK_RECTYP_VOL_MOU
```



```
: 1649      2936 2 THEN
: 1650      2937 2
: 1651      2938 2      Record subtype is mount... format the volume UIC, volume protection, and
: 1652      2939 2      volume mount flag message lines.
: 1653      2940 2
: 1654      2941 2      BEGIN
: 1655      2942 2      STATUS = PUT_PKT_VOLUIC(.RECORD_PTR);
: 1656      2943 2      IF NOT .STATUS THEN RETURN .STATUS;
: 1657      2944 2      PUT_PKT_VOLPRO(.RECORD_PTR);
: 1658      2945 2      PUT_PKT_MOUFLG(.RECORD_PTR);
: 1659      2946 2      END
: 1660      2947 2 ELSE
: 1661      2948 2
: 1662      2949 2      Record subtype is dismount... format the dismount flag message lines.
: 1663      2950 2
: 1664      2951 2      PUT_PKT_DMOUFLG(.RECORD_PTR);
: 1665      2952 2      RETURN $$$_NORMAL;
: 1666      2953 1 END;
```

! End of FORMAT_VOL

```
                                .PSECT $PLIT$,NOWRT,NOEXE,2
6E 75 6F 6D 20 65 6D 75 6C 6F 56 20 2F 20 0F 004E7 .BLKB 1
                                004E8 P.ABZ: .ASCII <15>\ / Volume mount\
6D 73 69 64 20 65 6D 75 6C 6F 56 20 2F 20 12 004F7
                                004F8 P.ACA: .ASCII <18>\ / Volume dismount\
                                00507
```

```
                                .PSECT $OWN$,NOEXE,2
00000000' 00000000' 00BA0 BANNER_VEC:
                                .ADDRESS P.ABZ, P.ACA
```

```
                                .PSECT $CODE$,NOWRT,2
                                000C 00000 FORMAT_VOL:
                                .WORD Save R2,R3
                                52 04 AC D0 00002 .MOVL RECORD_PTR, R2
                                50 02 A2 3C 00006 .MOVZWL 2(R2), R0
                                0000'CF40 DD 0000A .PUSHL BANNER_VEC-4[R0]
                                52 DD 0000F .PUSHL R2
0000V CF 02 FB 00011 .CALLS #2, PUT_COMMON
                                53 D0 00016 .MOVL R0, STATUS
                                3C 53 E9 00019 .BLBC STATUS, 2$
                                52 DD 0001C .PUSHL R2
0000V CF 01 FB 0001E .CALLS #1, PUT_PKT_IMGNAME
                                52 DD 00023 .PUSHL R2
0000V CF 01 FB 00025 .CALLS #1, PUT_DEVNAME_LINE
                                52 DD 0002A .PUSHL R2
0000V CF 01 FB 0002C .CALLS #1, PUT_PKT_LOGNAME
                                52 DD 00031 .PUSHL R2
0000V CF 01 FB 00033 .CALLS #1, PUT_PKT_VOLNAME
                                0000' CF 95 00038 .TSTB PKT_COUNT_TAB+7
                                07 13 0003C .BEQL 1$
```


0000V	CF		52	DD	0003E		PUSHL	R2		:	2934
	01		01	FB	00040		CALLS	#1, PUT_PKT_VOLSNAME		:	
		02	A2	B1	00045	1\$:	CMPW	2(R2), #1		:	2935
			21	12	00049		BNEQ	4\$		:	
			52	DD	0004B		PUSHL	R2		:	2942
0000V	CF		01	FB	0004D		CALLS	#1, PUT_PKT_VOLUIC		:	
	53		50	DO	00052		MOVL	R0, STATUS		:	
	04		53	E8	00055		BLBS	STATUS, 3\$		:	2943
	50		53	DO	00058	2\$:	MOVL	STATUS, R0		:	
				04	0005B		RET			:	
			52	DD	0005C	3\$:	PUSHL	R2		:	2944
0000V	CF		01	FB	0005E		CALLS	#1, PUT_PKT_VOLPRO		:	
			52	DD	00063		PUSHL	R2		:	2945
0000V	CF		01	FB	00065		CALLS	#1, PUT_PKT_MOUFLG		:	
			07	11	0006A		BRB	5\$		:	2935
			52	DD	0006C	4\$:	PUSHL	R2		:	2951
0000V	CF		01	FB	0006E		CALLS	#1, PUT_PKT_DMOUFLG		:	
	50		01	DO	00073	5\$:	MOVL	#1, R0		:	2952
				04	00076		RET			:	2953

; Routine Size: 119 bytes, Routine Base: \$CODE\$ + 067A


```

: 1668 2954 1 ROUTINE PUT_COMMON(RECORD_PTR, ASCIC_PTR) =
: 1669 2955 1
: 1670 2956 1 ++
: 1671 2957 1 FUNCTIONAL DESCRIPTION:
: 1672 2958 1
: 1673 2959 1 This routine moves the message identification string into the alarm
: 1674 2960 1 buffer, and formats all of the common security alarm message lines and
: 1675 2961 1 puts them in the alarm buffer. The common information consists of the
: 1676 2962 1 event time, the process ID, and the process user name. This information
: 1677 2963 1 is obtained from the security alarm record header.
: 1678 2964 1
: 1679 2965 1 INPUT:
: 1680 2966 1 RECORD_PTR : The security alarm record address.
: 1681 2967 1 ASCIC_PTR : Address of the counted ASCII message identification
: 1682 2968 1 string.
: 1683 2969 1
: 1684 2970 1 IMPLICIT INPUT:
: 1685 2971 1 NONE.
: 1686 2972 1
: 1687 2973 1 OUTPUT:
: 1688 2974 1 NONE.
: 1689 2975 1
: 1690 2976 1 IMPLICIT OUTPUT:
: 1691 2977 1 Formatted common message lines in the alarm buffer.
: 1692 2978 1
: 1693 2979 1 SIDE EFFECTS:
: 1694 2980 1 NONE.
: 1695 2981 1
: 1696 2982 1 ROUTINE VALUE:
: 1697 2983 1 SS$_NORMAL or a $FA0 system service error.
: 1698 2984 1 --
: 1699 2985 1
: 1700 2986 2 BEGIN ! Start of PUT_COMMON
: 1701 2987 2
: 1702 2988 2 LITERAL
: 1703 2989 2 TIM_BUF_SIZ = 23;
: 1704 2990 2
: 1705 2991 2 MAP
: 1706 2992 2 RECORD_PTR : REF $BBLOCK;
: 1707 2993 2
: 1708 2994 2 BIND
: 1709 2995 2 TIME_BUF = UPLIT(BYTE(%ASCIC ' Time: ')),
: 1710 2996 2 PID_BUF = UPLIT(BYTE(%ASCIC ' PID: ')),
: 1711 2997 2 NAME_BUF = UPLIT(BYTE(%ASCIC ' User Name: ')),
: 1712 2998 2 CONTROL = %ASCID '!XL';
: 1713 2999 2
: 1714 3000 2 LOCAL
: 1715 3001 2 BUFFER : VECTOR(TIM_BUF_SIZ,BYTE),
: 1716 3002 2 TIME_DESCR : $BBLOCK[DSC$_S_BLN]
: 1717 3003 2 PRESET([DSC$_LENGTH] = TIM_BUF_SIZ,
: 1718 3004 2 [DSC$_DTYPE] = DSC$_DTYPE_T,
: 1719 3005 2 [DSC$_CLASS] = DSC$_CLASS_D,
: 1720 3006 2 [DSC$_POINTER] = BUFFER),
: 1721 3007 2 LENGTH : LONG,
: 1722 3008 2 STATUS : LONG;
: 1723 3009 2
: 1724 3010 2 !

```


	56	0000V	CF	9E	00002	.WORD	Save R2,R3,R4,R5,R6	2954
	55	0000'	CF	9E	00007	MOVAB	PUT_STRING_ASCII, R6	
	54	0000V	CF	9E	0000C	MOVAB	ALARM_DESCR, R5	
	5E		24	C2	00011	MOVAB	PUT_NEWLINE, R4	
04	AE	020E0017	8F	D0	00014	SUBL2	#36, SP	
08	AE	0C	AE	9E	0001C	MOVL	#34471959, TIME_DESCR	3006
		08	AC	DD	00021	MOVAB	BUFFER, TIME_DESCR+4	
	66		01	FB	00024	PUSHL	ASCII_PTR	3013
			7E	D4	00027	CALLS	#1, PUT_STRING_ASCII	
	64		01	FB	00029	CLRL	-(SP)	3014
			7E	D4	0002C	CALLS	#1, PUT_NEWLINE	
	52	04	AC	D0	0002E	CLRL	-(SP)	3020
		10	A2	9F	00032	MOVL	RECORD_PTR, R2	
		0C	AE	9F	00035	PUSHAB	16(R2)	
		10	AE	9F	00038	PUSHAB	TIME_DESCR	
00000000G	00		04	FB	0003B	PUSHAB	TIME_DESCR	
	53		50	D0	00042	CALLS	#4, SYS\$ASCTIM	
	3E		53	E9	00045	MOVL	R0, STATUS	
		0000'	CF	9F	00048	BLBC	STATUS, 2\$	3021
	66		01	FB	0004C	PUSHAB	TIME_BUF	3022
	20	0C	AE	91	0004F	CALLS	#1, PUT_STRING_ASCII	
			04	12	00053	CMPB	BUFFER, #32	3023
			30	90	00055	BNEQ	1\$	
0C	AE		AE	9F	00059	MOVB	#48, BUFFER	
		04	AE	9F	00059	PUSHAB	TIME_DESCR	3024
0000V	CF		01	FB	0005C	CALLS	#1, PUT_STRING_ASCII	
			7E	D4	00061	CLRL	-(SP)	3025
	64		01	FB	00063	CALLS	#1, PUT_NEWLINE	
		0000'	CF	9F	00066	PUSHAB	PID_BUF	3029
	66		01	FB	0006A	CALLS	#1, PUT_STRING_ASCII	
		0C	A2	DD	0006D	PUSHL	12(R2)	3030
			55	DD	00070	PUSHL	R5	
		08	AE	9F	00072	PUSHAB	LENGTH	
		0000'	CF	9F	00075	PUSHAB	CONTROL	
00000000G	00		04	FB	00079	CALLS	#4, SYS\$FAO	
	53		50	D0	00080	MOVL	R0, STATUS	
	04		53	E8	00083	BLBS	STATUS, 3\$	3031
	50		53	D0	00086	MOVL	STATUS, R0	
			04	00089	RET			
	65		6E	A2	0008A	SUBW2	LENGTH, ALARM_DESCR	3032
	50		6E	3C	0008D	MOVZWL	LENGTH, R0	3033
04	A5		50	C0	00090	ADDL2	R0, ALARM_DESCR+4	
			7E	D4	00094	CLRL	-(SP)	3034
	64		01	FB	00096	CALLS	#1, PUT_NEWLINE	
		0000'	CF	9F	00099	PUSHAB	NAME_BUF	3038
	66		01	FB	0009D	CALLS	#1, PUT_STRING_ASCII	
		38	A2	9F	000A0	PUSHAB	56(R2)	3039
			0C	DD	000A3	PUSHL	#12	
0000V	CF		02	FB	000A5	CALLS	#2, PUT_STRING	
			55	DD	000AA	PUSHL	R5	3040
0000V	CF		01	FB	000AC	CALLS	#1, STRIP_BLANKS_TRAIL	
			7E	D4	000B1	CLRL	-(SP)	3041
	64		01	FB	000B3	CALLS	#1, PUT_NEWLINE	
	50		01	D0	000B6	MOVL	#1, R0	3042
			04	000B9	RET			3043

; Routine Size: 186 bytes, Routine Base: \$CODE\$ + 06F1

OPCSSECURITY
V04-000

M 1
16-Sep-1984 01:48:50
14-Sep-1984 12:50:55

VAX-11 Bliss-32 V4.0-742
[OPCOM.SRC]SECURITY.B32;1

Page 60
(14)


```

: 1759 3044 1 ROUTINE PUT_NETUAF_LINE(RECORD_PTR, RECNUM) : NOVALUE =
: 1760 3045 1
: 1761 3046 1 |++
: 1762 3047 1 | FUNCTIONAL DESCRIPTION:
: 1763 3048 1 |
: 1764 3049 1 | This routine formats a network authorization line. A network
: 1765 3050 1 | authorization line consists of the remote node and user name, and the
: 1766 3051 1 | local name the user is mapped to.
: 1767 3052 1 |
: 1768 3053 1 | INPUT:
: 1769 3054 1 | RECORD_PTR : The security alarm record address.
: 1770 3055 1 | RECNUM : Specifies which NODENAM and USERNAM packets to use.
: 1771 3056 1 |
: 1772 3057 1 | IMPLICIT INPUT:
: 1773 3058 1 | NONE.
: 1774 3059 1 |
: 1775 3060 1 | OUTPUT:
: 1776 3061 1 | NONE.
: 1777 3062 1 |
: 1778 3063 1 | IMPLICIT OUTPUT:
: 1779 3064 1 | Formatted network authorization line in the alarm buffer.
: 1780 3065 1 |
: 1781 3066 1 | SIDE EFFECTS:
: 1782 3067 1 | NONE.
: 1783 3068 1 |
: 1784 3069 1 | ROUTINE VALUE:
: 1785 3070 1 | NONE.
: 1786 3071 1 | --
: 1787 3072 1 |
: 1788 3073 2 BEGIN ! Start of PUT_NETUAF_LINE
: 1789 3074 2
: 1790 3075 2 MAP
: 1791 3076 2 RECORD_PTR : REF $BBLOCK;
: 1792 3077 2
: 1793 3078 2 BIND
: 1794 3079 2 BLANKS = UPLIT(BYTE(%ASCII ' '));
: 1795 3080 2
: 1796 3081 2 LOCAL
: 1797 3082 2 PTR : LONG;
: 1798 3083 2
: 1799 3084 2 PTR = .ALARM_DESCR[DSC$A_POINTER] + 6;
: 1800 3085 2 PUT_PKT_STRING(.RECORD_PTR, NS$K_PKTTYP_NODENAM, .RECNUM);
: 1801 3086 2 IF .ALARM_DESCR[DSC$A_POINTER] NEQU .PTR
: 1802 3087 2 THEN
: 1803 3088 2 PUT_STRING(.PTR - .ALARM_DESCR[DSC$A_POINTER], BLANKS);
: 1804 3089 2 PUT_STRING_ASCII(COLONS_BUF);
: 1805 3090 2 PTR = .ALARM_DESCR[DSC$A_POINTER] + 13;
: 1806 3091 2 PUT_PKT_STRING(.RECORD_PTR, NS$K_PKTTYP_USERNAM, (.RECNUM-1)*2+1);
: 1807 3092 2 PUT_STRING(.PTR - .ALARM_DESCR[DSC$A_POINTER], BLANKS);
: 1808 3093 2 PUT_PKT_STRING(.RECORD_PTR, NS$K_PKTTYP_USERNAM, (.RECNUM-1)*2+2);
: 1809 3094 2 PUT_NEWLINE(0);
: 1810 3095 1 END; ! End of PUT_NETUAF_LINE

```

.PSECT \$PLITS,NOWRT,NOEXE,2

20 20 20 20 20 20 20 20 20 20 20 20 0053C P.ACG: .ASCII \

BLANKS=

P.ACG

.PSECT \$CODE\$,NOWRT,2

003C 00000 PUT_NETUAF LINE:

	55	0000V	CF	9E	00002	.WORD	Save R2,R3,R4,R5	: 3044
	54	0000'	CF	9E	00007	MOVAB	PUT_PKT_STRING, R5	:
53	64		06	C1	0000C	MOVAB	ALARM_DESCR+4, R4	:
		08	AC	DD	00010	ADDL3	#6, ALARM_DESCR+4, PTR	: 3084
			09	DD	00013	PUSHL	RECNUM	: 3085
		04	AC	DD	00015	PUSHL	#9	:
	65		03	FB	00018	PUSHL	RECORD_PTR	:
	53		64	D1	0001B	CALLS	#3, PUT_PKT_STRING	:
			0D	13	0001E	CMPL	ALARM_DESCR+4, PTR	: 3086
		0000'	CF	9F	00020	BEQL	1\$	:
7E			64	C3	00024	PUSHAB	BLANKS	: 3088
	0000V	CF	02	FB	00028	SUBL3	ALARM_DESCR+4, PTR, -(SP)	:
		0000'	CF	9F	0002D	CALLS	#2, PUT_STRING	:
			01	FB	00031	PUSHAB	COLONS_BUF	: 3089
53	0000V	CF	0D	C1	00036	CALLS	#1, PUT_STRING_ASCII	:
			06	D7	00042	ADDL3	#13, ALARM_DESCR+4, PTR	: 3090
		08	AC	D0	0003A	MOVL	RECNUM, R2	: 3091
7E			01	78	0003E	ASHL	#1, R2, -(SP)	:
			0A	DD	00044	DECL	(SP)	:
		04	AC	DD	00046	PUSHL	#10	:
			03	FB	00049	PUSHL	RECORD_PTR	:
	65		CF	9F	0004C	CALLS	#3, PUT_PKT_STRING	:
7E		0000'	64	C3	00050	PUSHAB	BLANKS	: 3092
	0000V	CF	02	FB	00054	SUBL3	ALARM_DESCR+4, PTR, -(SP)	:
7E			01	78	00059	CALLS	#2, PUT_STRING	:
			0A	DD	0005D	ASHL	#1, R2, -(SP)	: 3093
		04	AC	DD	0005F	PUSHL	#10	:
			03	FB	00062	PUSHL	RECORD_PTR	:
	65		7E	D4	00065	CALLS	#3, PUT_PKT_STRING	:
	0000V	CF	01	FB	00067	CLRL	-(SP)	: 3094
			04	0006C	CALLS	#1, PUT_NEWLINE	:	
					RET			: 3095

; Routine Size: 109 bytes, Routine Base: \$CODE\$ + 07AB


```

: 1812 3096 1 ROUTINE PUT_DEVNAM_LINE(RECORD_PTR) : NOVALUE =
: 1813 3097 1
: 1814 3098 1 ++
: 1815 3099 1 FUNCTIONAL DESCRIPTION:
: 1816 3100 1
: 1817 3101 1 This routine formats a device name line and puts it in the security
: 1818 3102 1 alarm buffer. A device name line consists of the device name, (obtained
: 1819 3103 1 from the first DEVNAM packet), and optionally a second device name in
: 1820 3104 1 parenthesis if there is a second DEVNAM packet.
: 1821 3105 1
: 1822 3106 1 INPUT:
: 1823 3107 1 RECORD_PTR : The security alarm record address.
: 1824 3108 1
: 1825 3109 1 IMPLICIT INPUT:
: 1826 3110 1 NONE.
: 1827 3111 1
: 1828 3112 1 OUTPUT:
: 1829 3113 1 NONE.
: 1830 3114 1
: 1831 3115 1 IMPLICIT OUTPUT:
: 1832 3116 1 Formatted device name line in the alarm buffer.
: 1833 3117 1
: 1834 3118 1 SIDE EFFECTS:
: 1835 3119 1 NONE.
: 1836 3120 1
: 1837 3121 1 ROUTINE VALUE:
: 1838 3122 1 NONE.
: 1839 3123 1 --
: 1840 3124 1
: 1841 3125 2 BEGIN ! Start of PUT_DEVNAM_LINE
: 1842 3126 2
: 1843 3127 2 MAP
: 1844 3128 2 RECORD_PTR : REF $BLOCK;
: 1845 3129 2
: 1846 3130 2 BIND
: 1847 3131 2 DEVNAM_BUF = UPLIT(BYTE(%ASCIC ' Dev Name: ')),
: 1848 3132 2 LEFT_BUF = UPLIT(BYTE(%ASCIC ' (')),
: 1849 3133 2 RIGHT_BUF = UPLIT(BYTE(%ASCIC ' '));
: 1850 3134 2
: 1851 3135 2
: 1852 3136 2 ! Format the first part of the device name message line.
: 1853 3137 2
: 1854 3138 2 PUT_STRING_ASCII(DEVNAM_BUF);
: 1855 3139 2 PUT_PKT_STRING(.RECORD_PTR, NSAK_PKTTYPE_DEVNAM, 1);
: 1856 3140 2
: 1857 3141 2 ! If there is a second device name packet, append it within parenthesis to
: 1858 3142 2 the device name message line.
: 1859 3143 2
: 1860 3144 2 IF .PKT_COUNT_TAB[NSAK_PKTTYPE_DEVNAM-1] EQLU 2
: 1861 3145 2 THEN
: 1862 3146 3 BEGIN
: 1863 3147 3 PUT_STRING_ASCII(LEFT_BUF);
: 1864 3148 3 PUT_PKT_STRING(.RECORD_PTR, NSAK_PKTTYPE_DEVNAM, 2);
: 1865 3149 3 PUT_STRING_ASCII(RIGHT_BUF);
: 1866 3150 3 END;
: 1867 3151 2 PUT_NEWLINE(0);
: 1868 3152 1 END; ! End of PUT_DEVNAM_LINE

```



```

                                .PSECT $SPLITS,NOWRT,NOEXE,2
09 3A 65 6D 61 4E 20 76 65 44 09 0B 00548 P.ACH: .ASCII <11><9>\Dev Name:\<9>
28 20 20 03 00554 P.ACI: .ASCII <3>\ (\
29 01 00558 P.ACJ: .ASCII <1>\)\

```

```

DEVNAM_BUF= P.ACH
LEFT_BUF= P.ACI
RIGHT_BUF= P.ACJ

```

```

                                .PSECT $CODE$,NOWRT,2
                                0004 00000 PUT_DEVNAM LINE:
                                .WORD Save R2
52 0000V CF 9E 00002 MOVAB PUT_STRING_ASCIC, R2
0000' CF 9F 00007 PUSHAB DEVNAM_BUF
62 01 FB 0000B CALLS #1, PUT_STRING_ASCIC
01 DD 0000E PUSHL #1
05 DD 00010 PUSHL #5
04 AC DD 00012 PUSHL RECORD_PTR
0000V CF 03 FB 00015 CALLS #3, PUT_PKT_STRING
02 0000' CF 91 0001A CMPB PKT_COUNT_TAB+4, #2
1A 12 0001F BNEQ 1$
0000' CF 9F 00021 PUSHAB LEFT_BUF
62 01 FB 00025 CALLS #1, PUT_STRING_ASCIC
02 DD 00028 PUSHL #2
05 DD 0002A PUSHL #5
04 AC DD 0002C PUSHL RECORD_PTR
0000V CF 03 FB 0002F CALLS #3, PUT_PKT_STRING
0000' CF 9F 00034 PUSHAB RIGHT_BUF
62 01 FB 00038 CALLS #1, PUT_STRING_ASCIC
7E D4 0003B 1$: CLRL -(SP)
0000V CF 01 FB 0003D CALLS #1, PUT_NEWLINE
04 00042 RET

```

; Routine Size: 67 bytes, Routine Base: \$CODE\$ + 0818


```

: 1870 3153 1 ROUTINE PUT_SOURCE_LINE(RECORD_PTR) =
: 1871 3154 1
: 1872 3155 1 ++
: 1873 3156 1 FUNCTIONAL DESCRIPTION:
: 1874 3157 1
: 1875 3158 1 This routine formats a network source line. A network source line
: 1876 3159 1 consists of the remote node number, node name, and the remote username.
: 1877 3160 1 This information is obtained from the first NODEID, NODENAM, and USERNAM
: 1878 3161 1 packets in the security auditing record.
: 1879 3162 1
: 1880 3163 1 INPUT:
: 1881 3164 1 RECORD_PTR : The security alarm record address.
: 1882 3165 1
: 1883 3166 1 IMPLICIT INPUT:
: 1884 3167 1 NONE.
: 1885 3168 1
: 1886 3169 1 OUTPUT:
: 1887 3170 1 NONE.
: 1888 3171 1
: 1889 3172 1 IMPLICIT OUTPUT:
: 1890 3173 1 Formatted source line in the alarm buffer.
: 1891 3174 1
: 1892 3175 1 SIDE EFFECTS:
: 1893 3176 1 NONE.
: 1894 3177 1
: 1895 3178 1 ROUTINE VALUE:
: 1896 3179 1 SSS_NORMAL or a $FAO system service error.
: 1897 3180 1 --
: 1898 3181 1
: 1899 3182 2 BEGIN ! Start of PUT_SOURCE_LINE
: 1900 3183 2
: 1901 3184 2 MAP
: 1902 3185 2 RECORD_PTR : REF $BBLOCK;
: 1903 3186 2
: 1904 3187 2 LITERAL
: 1905 3188 2 NODE_BUF_SIZE = 9;
: 1906 3189 2
: 1907 3190 2 BIND
: 1908 3191 2 SOURCE_BUF = UPLIT(BYTE(%ASCIC ' Source: ')),
: 1909 3192 2 CONTROL1 = %ASCID '!UB.!UW'
: 1910 3193 2 CONTROL2 = %ASCID '!+!UW';
: 1911 3194 2
: 1912 3195 2 LOCAL
: 1913 3196 2 BUFFER : VECTOR[NODE_BUF_SIZE,BYTE],
: 1914 3197 2 NODE_DESCR : $BBLOCK[DSC$C_S-BLN]
: 1915 3198 2 PRESET([DSC$W_LENGTH] = NODE_BUF_SIZE,
: 1916 3199 2 [DSC$B_DTYPE] = DSC$R_DTYPE_T,
: 1917 3200 2 [DSC$B_CLASS] = DSC$K_CLASS_D,
: 1918 3201 2 [DSC$A_POINTER] = BUFFER),
: 1919 3202 2 PKTPTR : REF $BBLOCK,
: 1920 3203 2 CONTROL : LONG,
: 1921 3204 2 STATUS : LONG;
: 1922 3205 2
: 1923 3206 2 !
: 1924 3207 2 Move the label field into the alarm buffer and form a pointer to the node
: 1925 3208 2 id packet.
: 1926 3209 2 !

```



```
: 1927      3210 2 PUT STRING ASCIC(SOURCE BUF);
: 1928      3211 2 PKTPTR = .PKT_POINTER_TAB[.PKT_INDEX_TAB[NSASK_PKTTPY_NODEID-1]];
: 1929      3212 2
: 1930      3213 2 Convert the node id and move it into the alarm buffer.
: 1931      3214 2
: 1932      3215 2 IF .PKTPTR[NMASV_AREA] NEQU 0
: 1933      3216 2 THEN
: 1934      3217 2 CONTROL = CONTROL1
: 1935      3218 2 ELSE
: 1936      3219 2 CONTROL = CONTROL2;
: 1937      3220 2 STATUS = $FAO(.CONTROL, NODE_DESCR[DSCSW_LENGTH], NODE_DESCR, .PKTPTR[NMASV_AREA], .PKTPTR[NMASV_ADDR]);
: 1938      3221 2 IF NOT .STATUS THEN RETURN .STATUS;
: 1939      3222 2 STRIP BLANKS STRING(NODE_DESCR);
: 1940      3223 2 PUT_STRING_ASCIC(NODE_DESCR);
: 1941      3224 2
: 1942      3225 2 Add " nodename::remoteid" to alarm buffer.
: 1943      3226 2 Substitute "(None)" for a 0 length node name.
: 1944      3227 2
: 1945      3228 2 PUT STRING ASCIC(SPACES BUF);
: 1946      3229 2 PKTPTR = .PKT_POINTER_TAB[.PKT_INDEX_TAB[NSASK_PKTTPY_NODENAM-1]];
: 1947      3230 2 IF (.PKTPTR[NSASW_PKT_SIZE] - NSASK_PKTHDR_LENGTH) GTRU 0
: 1948      3231 2 THEN
: 1949      3232 2 PUT_PKT_STRING(.RECORD_PTR, NSASK_PKTTPY_NODENAM, 1)
: 1950      3233 2 ELSE
: 1951      3234 2 PUT_STRING_ASCIC(NONE BUF);
: 1952      3235 2 PUT_STRING_ASCIC(COLONS BUF);
: 1953      3236 2 PUT_PKT_STRING(.RECORD_PTR, NSASK_PKTTPY_USERNAM, 1);
: 1954      3237 2 PUT_NEWLINE(0);
: 1955      3238 2 RETURN SSS_NORMAL;
: 1956      3239 1 END;
```

! End of PUT_SOURCE_LINE

```
.PSECT $PLIT$,NOWRT,NOEXE,2
09 09 3A 65 63 72 75 6F 53 09 0A 0055A .BLKB 2
00 57 55 21 2E 42 55 21 0055C P.ACK: .ASCII <10><9>\Source:\<9><9>
010E0007 00567 .BLKB 1
00000000' 00568 P.ACM: .ASCII \!UB.!UW\<0>
00 00 00 57 55 21 2B 21 00570 P.ACL: .LONG 17694727
010E0005' 00574 .ADDRESS P.ACM
00000000' 00578 P.ACO: .ASCII \!+!UW\<0><0><0>
00580 P.ACN: .LONG 17694725
00584 .ADDRESS P.ACO
```

```
SOURCE_BUF= P.ACK
CONTROL1= P.ACL
CONTROL2= P.ACN
```

.PSECT \$CODE\$,NOWRT,2

```
001C 0000 PUT_SOURCE_LINE:
54 0000' CF 9E 00002 .WORD Save R2,R3,R4
53 0000V CF 9E 00007 MOVAB PKT_POINTER_TAB, R4
5E 10 C2 0000C SUBL2 PUT_STRING_ASCIC, R3
#16, SP
```

3153

		020E0009	8F	DD	0000F	PUSHL	#34471945	: 3201
04	AE	08	AE	9E	00015	MOVAB	BUFFER, NODE_DESCR+4	: 3210
		0000'	CF	9F	0001A	PUSHAB	SOURCE_BUF	: 3211
	63		01	FB	0001E	CALLS	#1, PUT_STRING_ASCIC	: 3215
	50	F6	A4	3C	00021	MOVZWL	PKT_INDEX TAB+30, R0	: 3217
	52		6440	D0	00025	MOVL	PKT-POINTER TAB[R0], PKTPTR	: 3219
FC	8F	01	A2	93	00029	BITB	1(PKTPTR), #252	: 3220
			07	13	0002E	BEQL	1\$	: 3221
	50	0000'	CF	9E	00030	MOVAB	CONTROL1, CONTROL	: 3222
			05	11	00035	BRB	2\$	: 3223
	50	0000'	CF	9E	00037	MOVAB	CONTROL2, CONTROL	: 3228
7E	0A		00	EF	0003C	EXTZV	#0, #10, (PKTPTR), -(SP)	: 3229
7E	62	06	0A	EF	00041	EXTZV	#10, #6, (PKTPTR), -(SP)	: 3230
			08	AE	9F	PUSHAB	NODE_DESCR	: 3231
			0C	AE	9F	PUSHAB	NODE_DESCR	: 3232
			50	DD	0004C	PUSHL	CONTROL	: 3233
00000000G	00		05	FB	0004E	CALLS	#5, SYSSFA0	: 3234
	58		50	E9	00055	BLBC	STATUS, 5\$	: 3235
			5E	DD	00058	PUSHL	SP	: 3236
0000V	CF		01	FB	0005A	CALLS	#1, STRIP_BLANKS_STRING	: 3237
			5E	DD	0005F	PUSHL	SP	: 3238
0000V	CF		01	FB	00061	CALLS	#1, PUT_STRING_ASCIC	: 3239
		0000'	CF	9F	00066	PUSHAB	SPACES_BUF	: 3240
	63		01	FB	0006A	CALLS	#1, PUT_STRING_ASCIC	: 3241
	50	EB	A4	3C	0006D	MOVZWL	PKT_INDEX TAB+16, R0	: 3242
	52		6440	D0	00071	MOVL	PKT-POINTER TAB[R0], PKTPTR	: 3243
	50	FE	A2	3C	00075	MOVZWL	-2(PKTPTR), R0	: 3244
	50		04	C2	00079	SUBL2	#4, R0	: 3245
			0E	13	0007C	BEQL	3\$	: 3246
			01	DD	0007E	PUSHL	#1	: 3247
			09	DD	00080	PUSHL	#9	: 3248
		04	AC	DD	00082	PUSHL	RECORD_PTR	: 3249
0000V	CF		03	FB	00085	CALLS	#3, PUT_PKT_STRING	: 3250
			07	11	0008A	BRB	4\$	: 3251
		0000'	CF	9F	0008C	PUSHAB	NONE_BUF	: 3252
	63		01	FB	00090	CALLS	#1, PUT_STRING_ASCIC	: 3253
		0000'	CF	9F	00093	PUSHAB	COLONS_BUF	: 3254
	63		01	FB	00097	CALLS	#1, PUT_STRING_ASCIC	: 3255
			01	DD	0009A	PUSHL	#1	: 3256
			0A	DD	0009C	PUSHL	#10	: 3257
		04	AC	DD	0009E	PUSHL	RECORD_PTR	: 3258
0000V	CF		03	FB	000A1	CALLS	#3, PUT_PKT_STRING	: 3259
			7E	D4	000A6	CLRL	-(SP)	: 3260
0000V	CF		01	FB	000A8	CALLS	#1, PUT_NEWLINE	: 3261
	50		01	D0	000AD	MOVL	#1, R0	: 3262
			04	000B0	5\$:	RET		: 3263

; Routine Size: 177 bytes, Routine Base: \$CODE\$ + 085B


```

: 1958 3240 1 ROUTINE PUT_PKT_STATUS(RECORD_PTR) =
: 1959 3241 1
: 1960 3242 1 ++
: 1961 3243 1 FUNCTIONAL DESCRIPTION:
: 1962 3244 1
: 1963 3245 1 This routine gets the message associated with the status longword and
: 1964 3246 1 formats a status message line.
: 1965 3247 1
: 1966 3248 1 INPUT:
: 1967 3249 1 RECORD_PTR : The security alarm record address.
: 1968 3250 1
: 1969 3251 1 IMPLICIT INPUT:
: 1970 3252 1 NONE.
: 1971 3253 1
: 1972 3254 1 OUTPUT:
: 1973 3255 1 NONE.
: 1974 3256 1
: 1975 3257 1 IMPLICIT OUTPUT:
: 1976 3258 1 Formatted message line in the alarm buffer.
: 1977 3259 1
: 1978 3260 1 SIDE EFFECTS:
: 1979 3261 1 NONE.
: 1980 3262 1
: 1981 3263 1 ROUTINE VALUE:
: 1982 3264 1 $$$_NORMAL or a $GETMSG system service error.
: 1983 3265 1 --
: 1984 3266 1
: 1985 3267 2 BEGIN ! Start of PUT_PKT_STATUS
: 1986 3268 2
: 1987 3269 2 MAP
: 1988 3270 2 RECORD_PTR : REF $BBLOCK;
: 1989 3271 2
: 1990 3272 2 BIND
: 1991 3273 2 STATUS_BUF = UPLIT(BYTE(%ASCIC ' Status: '));
: 1992 3274 2
: 1993 3275 2 LOCAL
: 1994 3276 2 PKTPTR : REF $BBLOCK,
: 1995 3277 2 BUFF : VECTOR[256,BYTE],
: 1996 3278 2 BUFF_DESCR : $BBLOCK[DSC$S_BLN]
: 1997 3279 2 PRESET([DSC$W_LENGTH] = 256,
: 1998 3280 2 [DSC$B_DTYPE] = DSC$K_DTYPE_T,
: 1999 3281 2 [DSC$B_CLASS] = DSC$K_CLASS_D,
: 2000 3282 2 [DSC$A_POINTER] = BUFF),
: 2001 3283 2 STATUS : LONG;
: 2002 3284 2
: 2003 3285 2
: 2004 3286 2 Move the label field into the alarm buffer and form a pointer to the status
: 2005 3287 2 longword packet.
: 2006 3288 2
: 2007 3289 2 PUT_STRING_ASCII(STATUS_BUF);
: 2008 3290 2 PKTPTR = .PKT_POINTER_TAB[.PKT_INDEX_TAB[NSA$K_PKT_TYP_STATUS-1]];
: 2009 3291 2
: 2010 3292 2 Get the message text and move it into the message buffer.
: 2011 3293 2
: 2012 P 3294 2 STATUS = $GETMSG(MSGID = .PKTPTR[NSA$L_PKT_STATUS],
: 2013 P 3295 2 MSGLEN = BUFF_DESCR,
: 2014 P 3296 2 BUFADR = BUFF_DESCR,

```



```

: 2015      3297  2      FLAGS = 15);
: 2016      3298  2      IF NOT .STATUS THEN RETURN .STATUS;
: 2017      3299  2      WHILE .BUFF_DESCR[DSC$W_LENGTH] GTRU DATA_LINE_SIZE DO
: 2018      3300  3          BEGIN
: 2019      3301  3              PUT_STRING(DATA_LINE_SIZE, .BUFF_DESCR[DSC$A_POINTER]);
: 2020      3302  3              BUFF_DESCR[DSC$W_LENGTH] = .BUFF_DESCR[DSC$W_LENGTH] - DATA_LINE_SIZE;
: 2021      3303  3              BUFF_DESCR[DSC$A_POINTER] = .BUFF_DESCR[DSC$A_POINTER] + DATA_LINE_SIZE;
: 2022      3304  3              PUT_NEWLINE(1);
: 2023      3305  2          END;
: 2024      3306  2      PUT_STRING(.BUFF_DESCR[DSC$W_LENGTH], .BUFF_DESCR[DSC$A_POINTER]);
: 2025      3307  2      PUT_NEWLINE(0);
: 2026      3308  2      RETURN SS$NORMAL;
: 2027      3309  1      END;

```

! End of PUT_PKT_STATUS

.PSECT \$SPLITS,NOWRT,NOEXE,2

09 09 3A 73 75 74 61 74 53 09 0A 00588 P.ACP: .ASCII <10><9>\Status:\<9><9>

STATUS_BUF= P.ACP

.PSECT \$CODE\$,NOWRT,2

				0000 00000 PUT_PKT_STATUS:			
	SE	FEFC	CE	9E 00002	.WORD	Save nothing	: 3240
		020E0100	8F	DD 00007	MOVAB	-260(SP), SP	
04	AE	08	AE	9E 0000D	PUSHL	#34472192	: 3282
		0000'	CF	9F 00012	MOVAB	BUFF, BUFF_DESCR+4	
0000V	CF		01	FB 00016	PUSHAB	STATUS_BUF	: 3289
	50	0000'	CF	3C 0001B	CALLS	#1, PUT_STRING ASCII	
	50	0000'	CF	40 00020	MOVZWL	PKT_INDEX TAB+36, R0	: 3290
	7E		0F	7D 00026	MOVL	PKT_POINTER_TAB[R0], PKTPTR	
		08	AE	9F 00029	MOVQ	#15, -(SP)	: 3297
		0C	AE	9F 0002C	PUSHAB	BUFF_DESCR	
			60	DD 0002F	PUSHAB	BUFF_DESCR	
00000000G	00		05	FB 00031	PUSHL	(PKTPTR)	
	35		50	E9 00038	CALLS	#5, SYS\$GETMSG	
	37		6E	B1 0003B	BLBC	STATUS, 3\$	: 3298
			1A	1B 0003E	CMPW	BUFF_DESCR, #55	: 3299
		04	AE	DD 00040	BLEQU	2\$	
			37	DD 00043	PUSHL	BUFF_DESCR+4	: 3301
0000V	CF		02	FB 00045	PUSHL	#55	
	6E		37	A2 0004A	CALLS	#2, PUT_STRING	: 3302
04	AE		37	C0 0004D	SUBW2	#55, BUFF_DESCR	: 3303
			01	DD 00051	ADDL2	#55, BUFF_DESCR+4	: 3304
0000V	CF		01	FB 00053	PUSHL	#1	
			E1	11 00058	CALLS	#1, PUT_NEWLINE	: 3299
		04	AE	DD 0005A	BRB	1\$	: 3306
	7E	04	AE	3C 0005D	PUSHL	BUFF_DESCR+4	
0000V	CF		02	FB 00061	MOVZWL	BUFF_DESCR, -(SP)	
			7E	D4 00066	CALLS	#2, PUT_STRING	: 3307
0000V	CF		01	FB 00068	CLRL	-(SP)	
	50		01	D0 0006D	CALLS	#1, PUT_NEWLINE	: 3308
			04	00070	MOVL	#1, R0	: 3309
					RET		

OPC\$SECURITY
V04-000

J 2
16-Sep-1984 01:48:50
14-Sep-1984 12:50:55

VAX-11 Bliss-32 V4.0-742
[OPCOM.SRC]SECURITY.B32;1

Page 70
(18)

; Routine Size: 113 bytes, Routine Base: \$CODE\$ + 090C


```
2029 3310 1 ROUTINE PUT_PKT_PARENTPID(RECORD_PTR) =
2030 3311 1
2031 3312 1 !++
2032 3313 1 FUNCTIONAL DESCRIPTION:
2033 3314 1
2034 3315 1 This routine formats a security alarm message line from the
2035 3316 1 first EPID packet in the security auditing record.
2036 3317 1
2037 3318 1 INPUT:
2038 3319 1 RECORD_PTR : The security alarm record address.
2039 3320 1
2040 3321 1 IMPLICIT INPUT:
2041 3322 1 NONE.
2042 3323 1
2043 3324 1 OUTPUT:
2044 3325 1 NONE.
2045 3326 1
2046 3327 1 IMPLICIT OUTPUT:
2047 3328 1 Formatted message line in the alarm buffer.
2048 3329 1
2049 3330 1 SIDE EFFECTS:
2050 3331 1 NONE.
2051 3332 1
2052 3333 1 ROUTINE VALUE:
2053 3334 1 SSS_NORMAL or a $FA0 system service error.
2054 3335 1 --
2055 3336 1
2056 3337 2 BEGIN ! Start of PUT_PKT_PARENTPID
2057 3338 2
2058 3339 2 MAP
2059 3340 2 RECORD_PTR : REF $BBLOCK;
2060 3341 2
2061 3342 2 BIND
2062 3343 2 PID_BUF = UPLIT(BYTE(%ASCIC ' Parent PID: '),
2063 3344 2 CONTROL = %ASCID '!XL');
2064 3345 2
2065 3346 2 LOCAL
2066 3347 2 PKTPTR : REF $BBLOCK,
2067 3348 2 LENGTH : LONG,
2068 3349 2 STATUS : LONG;
2069 3350 2
2070 3351 2
2071 3352 2 Move the label field into the alarm buffer and form a pointer to the process
2072 3353 2 id packet.
2073 3354 2
2074 3355 2 PUT_STRING_ASCIC(PID_BUF);
2075 3356 2 PKTPTR = .PKT_POINTER_TAB[.PKT_INDEX_TAB[NSA$K_PKT_TYP_EPID-1]];
2076 3357 2
2077 3358 2 Convert and move the process id field into the message buffer.
2078 3359 2
2079 3360 2 STATUS = $FA0(CONTROL, LENGTH, ALARM_DESCR, .PKTPTR[NSA$L_PKT_EPID]);
2080 3361 2 IF NOT .STATUS THEN RETURN .STATUS;
2081 3362 2 ALARM_DESCR[DSC$W_LENGTH] = .ALARM_DESCR[DSC$W_LENGTH] - .LENGTH<0,16,0>;
2082 3363 2 ALARM_DESCR[DSC$A_POINTER] = .ALARM_DESCR[DSC$A_POINTER] + .LENGTH<0,16,0>;
2083 3364 2 PUT_NEWLINE(0);
2084 3365 2 RETURN SSS_NORMAL;
2085 3366 1 END; ! End of PUT_PKT_PARENTPID
```



```

                                .PSECT $PLITS$,NOWRT,NOEXE,2
09 3A 44 49 50 20 74 6E 65 72 61 50 09 0D 00593 .BLKB 1
                                .ASCII <13><9>\Parent PID:\<9>
                                00594 P.ACQ: .BLKB 2
                                005A2 .ASCII \!XL\<0>
00 4C 58 21 005A4 P.ACS: .LONG 17694723
                                010E0003 005A8 P.ACR: .ADDRESS P.ACS
                                00000000 005AC

```

PID_BUF= P.ACQ
CONTROL= P.ACR

```

                                .PSECT $CODE$,NOWRT,2
                                0004 0000 PUT_PKT_PARENTPID:
                                .WORD Save R2
                                MOVAB ALARM_DESCR, R2
                                SUBL2 #4, SP
                                PUSHAB PID_BUF
                                CALLS #1, PUT_STRING_ASCII
                                MOVZWL PKT_INDEX_TAB+32, R0
                                MOVL PKT_POINTER_TAB[R0], PKTPTR
                                PUSHL (PKTPTR)
                                PUSHL R2
                                PUSHAB LENGTH
                                PUSHAB CONTROL
                                CALLS #4, SY$FAO
                                BLBC STATUS, 1$
                                SUBW2 LENGTH, ALARM_DESCR
                                MOVZWL LENGTH, R0
                                ADDL2 R0, ALARM_DESCR+4
                                CLRL -(SP)
                                CALLS #1, PUT_NEWLINE
                                MOVL #1, R0
                                RET
                                04 00045 1$:

```

; Routine Size: 70 bytes, Routine Base: \$CODE\$ + 097D


```

: 2087 3367 1 ROUTINE PUT_PKT_PARENTUN(RECORD_PTR) : NOVALUE =
: 2088 3368 1
: 2089 3369 1 !++
: 2090 3370 1 FUNCTIONAL DESCRIPTION:
: 2091 3371 1
: 2092 3372 1 This routine formats a security alarm message line from the
: 2093 3373 1 first USERNAM packet in the security auditing record.
: 2094 3374 1
: 2095 3375 1 INPUT:
: 2096 3376 1 RECORD_PTR : The security alarm record address.
: 2097 3377 1
: 2098 3378 1 IMPLICIT INPUT:
: 2099 3379 1 NONE.
: 2100 3380 1
: 2101 3381 1 OUTPUT:
: 2102 3382 1 NONE.
: 2103 3383 1
: 2104 3384 1 IMPLICIT OUTPUT:
: 2105 3385 1 Formatted message line in the alarm buffer.
: 2106 3386 1
: 2107 3387 1 SIDE EFFECTS:
: 2108 3388 1 NONE.
: 2109 3389 1
: 2110 3390 1 ROUTINE VALUE:
: 2111 3391 1 NONE.
: 2112 3392 1 --
: 2113 3393 1
: 2114 3394 2 BEGIN ! Start of PUT_PKT_PARENTUN
: 2115 3395 2
: 2116 3396 2 MAP
: 2117 3397 2 RECORD_PTR : REF $BLOCK;
: 2118 3398 2
: 2119 3399 2 BIND
: 2120 3400 2 NAME_BUF = UPLIT(BYTE(%ASCIC ' Parent U.N.: '));
: 2121 3401 2
: 2122 3402 2 PUT_STRING_ASCIC(NAME_BUF);
: 2123 3403 2 PUT_PKT_STRING(.RECORD_PTR, NS$K_PKT_TYP_USERNAM, 1);
: 2124 3404 2 PUT_NEWLINE(0);
: 2125 3405 1 END; ! End of PUT_PKT_PARENTUN

```

```

                                .PSECT $PLIT$,NOWRT,NOEXE,2
09 3A 2E 4E 2E 55 20 74 6E 65 72 61 50 09 0E 005B0 P.ACT: .ASCII <14><9>\Parent U.N.:<9>
                                NAME_BUF= P.ACT

```

```

                                .PSECT $CODE$,NOWRT,2
                                0000 00000 PUT_PKT_PARENTUN:
                                .WORD Save nothing
                                PUSHAB NAME_BUF
                                CALLS #1, PUT_STRING_ASCIC
                                PUSHL #1
                                PUSHL #10
0000V CF 0000' 01 FB 00002
                                01 DD 0000B
                                0A DD 0000D

```

```

: 3367
: 3402
: 3403
:

```


OPC\$SECURITY
V04-000

N 2
16-Sep-1984 01:48:50
14-Sep-1984 12:50:55

VAX-11 Bliss-32 V4.0-742
[OPCOM.SRC]SECURITY.B32;1

Page 74
(20)

0000V	CF	04	AC	DD	0000F
			03	FB	00012
			7E	D4	00017
0000V	CF		01	FB	00019
			04		0001E

PUSHL	RECORD_PTR
CALLS	#3, PUT_PKT_STRING
CLRL	-(SP)
CALLS	#1, PUT_NEWLINE
RET	

:
:
: 3404
:
: 3405

; Routine Size: 31 bytes, Routine Base: \$CODE\$ + 09C3

```

: 2127      3406 1 ROUTINE PUT_PKT_VOLUIC(RECORD_PTR) =
: 2128      3407 1
: 2129      3408 1 !++
: 2130      3409 1 FUNCTIONAL DESCRIPTION:
: 2131      3410 1
: 2132      3411 1 This routine formats a security alarm message line from the
: 2133      3412 1 first UIC packet in the security auditing record.
: 2134      3413 1
: 2135      3414 1 INPUT:
: 2136      3415 1 RECORD_PTR : The security alarm record address.
: 2137      3416 1
: 2138      3417 1 IMPLICIT INPUT:
: 2139      3418 1 NONE.
: 2140      3419 1
: 2141      3420 1 OUTPUT:
: 2142      3421 1 NONE.
: 2143      3422 1
: 2144      3423 1 IMPLICIT OUTPUT:
: 2145      3424 1 Formatted message line in the alarm buffer.
: 2146      3425 1
: 2147      3426 1 SIDE EFFECTS:
: 2148      3427 1 NONE.
: 2149      3428 1
: 2150      3429 1 ROUTINE VALUE:
: 2151      3430 1 SSS_NORMAL or a $FAO system service error.
: 2152      3431 1 --
: 2153      3432 1
: 2154      3433 2 BEGIN ! Start of PUT_PKT_VOLUIC
: 2155      3434 2
: 2156      3435 2 MAP
: 2157      3436 2 RECORD_PTR : REF $BBLOCK;
: 2158      3437 2
: 2159      3438 2 BIND
: 2160      3439 2 VOLUIC_BUF = UPLIT(BYTE(%ASCIC ' Vol UIC: '),
: 2161      3440 2 CONTROL = %ASCID '!%I';
: 2162      3441 2
: 2163      3442 2 LOCAL
: 2164      3443 2 PKTPTR : REF $BBLOCK,
: 2165      3444 2 STATUS : LONG,
: 2166      3445 2 LENGTH : LONG;
: 2167      3446 2
: 2168      3447 2 !
: 2169      3448 2 Move the label field into the alarm buffer and form a pointer to the
: 2170      3449 2 volume uic packet.
: 2171      3450 2
: 2172      3451 2 PUT STRING ASCIC(VOLUIC_BUF);
: 2173      3452 2 PKTPTR = .PKT_POINTER_TAB[.PKT_INDEX_TAB[NSA$K_PKT_TYP_UIC-1]];
: 2174      3453 2
: 2175      3454 2 Convert and move the volume uic field into the message buffer.
: 2176      3455 2
: 2177      3456 2 STATUS = $FAO(CONTROL, LENGTH, ALARM_DESCR, .PKTPTR[NSA$L_PKT_UIC]);
: 2178      3457 2 IF NOT .STATUS THEN RETURN .STATUS;
: 2179      3458 2 ALARM_DESCR[DSC$W_LENGTH] = .ALARM_DESCR[DSC$W_LENGTH] - .LENGTH<0,16,0>;
: 2180      3459 2 ALARM_DESCR[DSC$A_POINTER] = .ALARM_DESCR[DSC$A_POINTER] + .LENGTH<0,16,0>;
: 2181      3460 2 PUT NEWLINE(0);
: 2182      3461 2 RETURN SSS_NORMAL;
: 2183      3462 1 END; ! End of PUT_PKT_VOLUIC
```


; Routine Size: 70 bytes, Routine Base: \$CODE\$ + 09E2

```
: 2185 3463 1 ROUTINE PUT_PKT_IMGNAM(RECORD_PTR) : NOVALUE =
: 2186 3464 1
: 2187 3465 1 ++
: 2188 3466 1 FUNCTIONAL DESCRIPTION:
: 2189 3467 1
: 2190 3468 1 This routine formats a security alarm message line from the
: 2191 3469 1 first IMGNAM packet in the security auditing record.
: 2192 3470 1
: 2193 3471 1 INPUT:
: 2194 3472 1 RECORD_PTR : The security alarm record address.
: 2195 3473 1
: 2196 3474 1 IMPLICIT INPUT:
: 2197 3475 1 NONE.
: 2198 3476 1
: 2199 3477 1 OUTPUT:
: 2200 3478 1 NONE.
: 2201 3479 1
: 2202 3480 1 IMPLICIT OUTPUT:
: 2203 3481 1 Formatted message line in the alarm buffer.
: 2204 3482 1
: 2205 3483 1 SIDE EFFECTS:
: 2206 3484 1 NONE.
: 2207 3485 1
: 2208 3486 1 ROUTINE VALUE:
: 2209 3487 1 NONE.
: 2210 3488 1 --
: 2211 3489 1
: 2212 3490 2 BEGIN ! Start of PUT_PKT_IMGNAM
: 2213 3491 2
: 2214 3492 2 MAP
: 2215 3493 2 RECORD_PTR : REF $BBLOCK;
: 2216 3494 2
: 2217 3495 2 BIND
: 2218 3496 2 IMGNAM_BUF = UPLIT(BYTE(%ASCIC ' Image: '));
: 2219 3497 2
: 2220 3498 2 LOCAL
: 2221 3499 2 PKTPTR : REF $BBLOCK;
: 2222 3500 2
: 2223 3501 2 PUT_STRING_ASCII(IMGNAM_BUF);
: 2224 3502 2 PKTPTR = .PKT_POINTER_TAB[.PKT_INDEX_TAB[NSASK_PKTTYP_IMGNAM-1]];
: 2225 3503 2 IF (.PKTPTR[NSASK_PKT_SIZE] - NSASK_PKTHDR_LENGTH) GTU 0
: 2226 3504 2 THEN
: 2227 3505 2 PUT_PKT_STRING(.RECORD_PTR, NSASK_PKTTYP_IMGNAM, 1)
: 2228 3506 2 ELSE
: 2229 3507 2 PUT_STRING_ASCII(NONE_BUF);
: 2230 3508 2 PUT_NEWLINE(0);
: 2231 3509 1 END; ! End of PUT_PKT_IMGNAM
```

.PSECT \$PLITS,NOWRT,NOEXE,2

09 09 3A 65 67 61 6D 49 09 09 005D8 P.ACX: .ASCII <9><9>\Image:\<9><9>

IMGNAM_BUF= P.ACX


```

.PSECT $CODE$,NOWRT,2
0000 00000 PUT_PKT_IMGNAM:
0000V CF 0000' CF 9F 00002 .WORD Save nothing : 3463
50 01 FB 00006 PUSHAB IMGNAM_BUF : 3501
50 0000' CF 3C 0000B CALLS #1, PUT_STRING_ASCII : 3502
50 0000' CF 40 D0 00010 MOVZWL PKT_INDEX_TAB, R0 : 3503
FE 04 3C 00016 MOVZWL PKT_POINTER_TAB[R0], PKTPTR : 3505
04 0E 13 0001D BEQL 1$ : 3507
01 DD 0001F PUSHL #1 : 3508
01 DD 00021 PUSHL #1 : 3509
04 AC DD 00023 PUSHL RECORD_PTR : 3507
0000V CF 03 FB 00026 CALLS #3, PUT_PKT_STRING : 3508
09 11 0002B BRB 2$ : 3509
0000V CF 0000' CF 9F 0002D 1$: PUSHAB NONE_BUF : 3507
01 FB 00031 CALLS #1, PUT_STRING_ASCII : 3508
7E D4 00036 2$: CLRL -(SP) : 3509
0000V CF 01 FB 00038 CALLS #1, PUT_NEWLINE : 3509
04 0003D RET : 3509

```

; Routine Size: 62 bytes, Routine Base: \$CODE\$ + 0A28

```
: 2233      3510 1 ROUTINE PUT_PKT_FILNAM(RECORD_PTR) : NOVALUE =
: 2234      3511 1
: 2235      3512 1 ++
: 2236      3513 1 FUNCTIONAL DESCRIPTION:
: 2237      3514 1
: 2238      3515 1 This routine formats a security alarm message line from the
: 2239      3516 1 first FILNAM packet in the security auditing record.
: 2240      3517 1
: 2241      3518 1 INPUT:
: 2242      3519 1 RECORD_PTR : The security alarm record address.
: 2243      3520 1
: 2244      3521 1 IMPLICIT INPUT:
: 2245      3522 1 NONE.
: 2246      3523 1
: 2247      3524 1 OUTPUT:
: 2248      3525 1 NONE.
: 2249      3526 1
: 2250      3527 1 IMPLICIT OUTPUT:
: 2251      3528 1 Formatted message line in the alarm buffer.
: 2252      3529 1
: 2253      3530 1 SIDE EFFECTS:
: 2254      3531 1 NONE.
: 2255      3532 1
: 2256      3533 1 ROUTINE VALUE:
: 2257      3534 1 NONE.
: 2258      3535 1 --
: 2259      3536 1
: 2260      3537 2 BEGIN ! Start of PUT_PKT_FILNAM
: 2261      3538 2
: 2262      3539 2 MAP
: 2263      3540 2 RECORD_PTR : REF $BLOCK;
: 2264      3541 2
: 2265      3542 2 BIND
: 2266      3543 2 FILNAM_BUF = UPLIT(BYTE(%ASCIC ' File: '));
: 2267      3544 2
: 2268      3545 2 PUT_STRING ASCIC(FILNAM_BUF);
: 2269      3546 2 PUT_PKT_STRING(.RECORD_PTR, NS$K_PKTTYP_FILNAM, 1);
: 2270      3547 2 PUT_NEWLINE(0);
: 2271      3548 1 END; ! End of PUT_PKT_FILNAM
```

```
.PSECT $PLITS,NOWRT,NOEXE,2
09 09 3A 65 6C 69 46 09 08 005E2 .BLKB 2
005E4 P.ACY: .ASCII <8><9>\File:\<9><9>
FILNAM_BUF= P.ACY
```

```
.PSECT $CODE$,NOWRT,2
0000 00000 PUT_PKT_FILNAM:
0000V CF 0000' CF 9F 00002 .WORD Save nothing : 3510
01 FB 00006 PUSHAB FILNAM_BUF : 3545
01 DD 0000B CALLS #1, PUT_STRING_ASCIC : 3546
PUSHL #1
```


OPC\$SECURITY
V04-000

G 3
16-Sep-1984 01:48:50
14-Sep-1984 12:50:55

VAX-11 Bliss-32 V4.0-742
[OPCOM.SRC]SECURITY.B32;1

Page 80
(23)

		04	DD	0000D	PUSHL	#4	
		AC	DD	0000F	PUSHL	RECORD_PTR	
0000V	CF	03	FB	00012	CALLS	#3, PUT_PKT_STRING	
		7E	D4	00017	CLRL	-(SP)	
0000V	CF	01	FB	00019	CALLS	#1, PUT_NEWLINE	
		04	0001E	RET			

:
:
:
: 3547
:
: 3548

; Routine Size: 31 bytes, Routine Base: \$CODE\$ + 0A66

```
: 2273 3549 1 ROUTINE PUT_PKT_FACMOD(RECORD_PTR) : NOVALUE =
: 2274 3550 1
: 2275 3551 1 !++
: 2276 3552 1 FUNCTIONAL DESCRIPTION:
: 2277 3553 1
: 2278 3554 1 This routine formats a security alarm message line from the
: 2279 3555 1 first FACMOD packet in the security auditing record.
: 2280 3556 1
: 2281 3557 1 INPUT:
: 2282 3558 1 RECORD_PTR : The security alarm record address.
: 2283 3559 1
: 2284 3560 1 IMPLICIT INPUT:
: 2285 3561 1 NONE.
: 2286 3562 1
: 2287 3563 1 OUTPUT:
: 2288 3564 1 NONE.
: 2289 3565 1
: 2290 3566 1 IMPLICIT OUTPUT:
: 2291 3567 1 Formatted message line in the alarm buffer.
: 2292 3568 1
: 2293 3569 1 SIDE EFFECTS:
: 2294 3570 1 NONE.
: 2295 3571 1
: 2296 3572 1 ROUTINE VALUE:
: 2297 3573 1 NONE.
: 2298 3574 1 --
: 2299 3575 1
: 2300 3576 2 BEGIN ! Start of PUT_PKT_FACMOD
: 2301 3577 2
: 2302 3578 2 LITERAL
: 2303 3579 2 NUM_FACMOD_BITS = 5;
: 2304 3580 2
: 2305 3581 2 MAP
: 2306 3582 2 RECORD_PTR : REF $BBLOCK;
: 2307 3583 2
: 2308 3584 2 BIND
: 2309 3585 2 FACMOD_BUF = UPLIT(BYTE(%ASCIC ' Mode: '));
: 2310 3586 2
: 2311 3587 2 OWN
: 2312 3588 2 KEYWORD_VEC : VECTOR[ NUM_FACMOD_BITS ]
: 2313 3589 2 INITIAL( UPLIT( BYTE(%ASCIC 'READ ')),
: 2314 3590 2 UPLIT( BYTE(%ASCIC 'WRITE ')),
: 2315 3591 2 UPLIT( BYTE(%ASCIC 'EXECUTE ')),
: 2316 3592 2 UPLIT( BYTE(%ASCIC 'DELETE ')),
: 2317 3593 2 UPLIT( BYTE(%ASCIC 'CONTROL ')));
: 2318 3594 2
: 2319 3595 2 LOCAL
: 2320 3596 2 PKTPTR : REF $BBLOCK;
: 2321 3597 2
: 2322 3598 2 PUT STRING ASCIC(FACMOD_BUF);
: 2323 3599 2 PKTPTR = .PKT POINTER TAB[.PKT INDEX TAB[NSASK_PKT_TYP FACMOD-1]];
: 2324 3600 2 PUT KEYWORDS(NUM_FACMOD_BITS, PKTPTR[NSASK_PKT_FACMOD], KEYWORD_VEC);
: 2325 3601 2 STRIP BLANKS TRAIL(ALARM_DESCR);
: 2326 3602 2 PUT_NEWLINE(0);
: 2327 3603 1 END; ! End of PUT_PKT_FACMOD
```



```

.PSECT $PLITS$,NOWRT,NOEXE,2
09 09 3A 65 64 6F 4D 09 08 005ED .BLKB 3
005F0 P.ACZ: .ASCII <8><9>\Mode:\<9><9> ;
005F9 .BLKB 3
20 44 41 45 52 05 005FC P.ADA: .ASCII <5>\READ \ ;
00602 .BLKB 2
20 45 54 49 52 57 06 00604 P.ADB: .ASCII <6>\WRITE \ ;
0060B .BLKB 1
20 45 54 55 43 45 58 45 08 0060C P.ADC: .ASCII <8>\EXECUTE \ ;
00615 .BLKB 3
20 45 54 45 4C 45 44 07 00618 P.ADD: .ASCII <7>\DELETE \ ;
4C 4F 52 54 4E 4F 43 07 00620 P.ADE: .ASCII <7>\CONTROL \ ;

```

```

.PSECT $OWNS$,NOEXE,2
00000000' 00000000' 00000000' 00000000' 00000000' 00BA8 KEYWORD_VEC:
- .ADDRESS P.ADA, P.ADB, P.ADC, P.ADD, P.ADE ;
FACMOD_BUF= P.ACZ

```

```

.PSECT $CODE$,NOWRT,2
0000 00000 PUT_PKT_FACMOD:
.WORD Save nothing : 3549
PUSHAB FACMOD_BUF : 3598
CALLS #1, PUT_STRING_ASCII :
MOVZWL PKT_INDEX_TAB+2, R0 : 3599
MOVL PKT_POINTER_TAB[R0], PKTPTR :
PUSHAB KEYWORD_VEC : 3600
PUSHL PKTPTR :
PUSHL #5 :
CALLS #3, PUT_KEYWORDS :
PUSHAB ALARM_DESCR : 3601
CALLS #1, STRIP_BLANKS_TRAIL :
CLRL -(SP) : 3602
CALLS #1, PUT_NEWLINE :
RET : 3603

```

; Routine Size: 52 bytes, Routine Base: \$CODE\$ + 0A85

```

: 2329 3604 1 ROUTINE PUT_PKT_PRIVUSED(RECORD_PTR) : NOVALUE =
: 2330 3605 1
: 2331 3606 1 ++
: 2332 3607 1 FUNCTIONAL DESCRIPTION:
: 2333 3608 1
: 2334 3609 1 This routine formats a security alarm message line from the
: 2335 3610 1 first PRIVUSED packet in the security auditing record.
: 2336 3611 1
: 2337 3612 1 INPUT:
: 2338 3613 1 RECORD_PTR : The security alarm record address.
: 2339 3614 1
: 2340 3615 1 IMPLICIT INPUT:
: 2341 3616 1 NONE.
: 2342 3617 1
: 2343 3618 1 OUTPUT:
: 2344 3619 1 NONE.
: 2345 3620 1
: 2346 3621 1 IMPLICIT OUTPUT:
: 2347 3622 1 Formatted message line in the alarm buffer.
: 2348 3623 1
: 2349 3624 1 SIDE EFFECTS:
: 2350 3625 1 NONE.
: 2351 3626 1
: 2352 3627 1 ROUTINE VALUE:
: 2353 3628 1 NONE.
: 2354 3629 1 --
: 2355 3630 1
: 2356 3631 2 BEGIN ! Start of PUT_PKT_PRIVUSED
: 2357 3632 2
: 2358 3633 2 LITERAL
: 2359 3634 2 NUM_PRIVUSED_BITS = 6;
: 2360 3635 2
: 2361 3636 2 MAP
: 2362 3637 2 RECORD_PTR : REF $BBLOCK;
: 2363 3638 2
: 2364 3639 2 BIND
: 2365 3640 2 PRIV_BUF = UPLIT(BYTE(%ASCIC ' Privs Used: '));
: 2366 3641 2
: 2367 3642 2 OWN
: 2368 3643 2 KEYWORD_VEC : VECTOR[ NUM PRIVUSED BITS]
: 2369 3644 2 INITIAL( UPLIT(BYTE(%ASCIC 'SYSPRV ')),
: 2370 3645 2 UPLIT(BYTE(%ASCIC 'BYPASS ')),
: 2371 3646 2 UPLIT(BYTE(%ASCIC 'UPGRADE ')),
: 2372 3647 2 UPLIT(BYTE(%ASCIC 'DOWNGRADE ')),
: 2373 3648 2 UPLIT(BYTE(%ASCIC 'GRPPRV ')),
: 2374 3649 2 UPLIT(BYTE(%ASCIC 'READALL')));
: 2375 3650 2
: 2376 3651 2 LOCAL
: 2377 3652 2 PKTPTR : REF $BBLOCK;
: 2378 3653 2
: 2379 3654 2 PUT_STRING_ASCII(PRIV_BUF);
: 2380 3655 2 PKTPTR = .PKT_POINTER_TAB[.PKT_INDEX_TAB[NSASK_PKT_TYP_PRIVUSED-1]];
: 2381 3656 2 IF .PKTPTR[NSA$L_PKT_PRIVUSED] NEQ 0
: 2382 3657 2 THEN
: 2383 3658 3 BEGIN
: 2384 3659 3 PUT_KEYWORDS(NUM PRIVUSED BITS, PKTPTR[NSA$L_PKT_PRIVUSED], KEYWORD_VEC);
: 2385 3660 3 STRIP_BLANKS_TRAIL(ALARM_DESCR);

```



```

: 2386      3661 3  END
: 2387      3662 2 ELSE
: 2388      3663 2 PUT_STRING_ASCII(NONE_BUF);
: 2389      3664 2 PUT_NEWLINE(0);
: 2390      3665 1 END;

```

! End of PUT_PKT_PRIVUSED

```

                                .PSECT $SPLIT$,NOWRT,NOEXE,2
09 3A 64 65 73 55 20 73 76 69 72 50 09 0D 00628 P.ADF: .ASCII <13><9>\Privs Used:\<9> ;
                                00636 .BLKB 2 ;
                                20 56 52 50 53 59 53 07 00638 P.ADG: .ASCII <7>\SYSPRV \ ;
                                20 53 53 41 50 59 42 07 00640 P.ADH: .ASCII <7>\BYPASS \ ;
                                20 45 44 41 52 47 50 55 08 00648 P.ADI: .ASCII <8>\UPGRADE \ ;
                                00651 .BLKB 3 ;
                                20 45 44 41 52 47 4E 57 4F 44 0A 00654 P.ADJ: .ASCII <10>\DOWNGRADE \ ;
                                0065F .BLKB 1 ;
                                20 56 52 50 50 52 47 07 00660 P.ADK: .ASCII <7>\GRPPRV \ ;
                                4C 4C 41 44 41 45 52 07 00668 P.ADL: .ASCII <7>\READALL\ ;

                                .PSECT $OWNS$,NOEXE,2
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00BBC KEYWORD_VEC:
                                .ADDRESS P.ADG, P.ADH, P.ADI, P.ADJ, P.ADK, P.ADL ;
                                PRIV_BUF= P.ADF

                                .PSECT $CODE$,NOWRT,2
                                0000 00000 PUT_PKT_PRIVUSED:
                                .WORD Save nothing ; 3604
                                PUSHAB PRIV_BUF ; 3654
                                CALLS #1, PUT_STRING_ASCII
                                MOVZWL PKT_INDEX_TAB+4, R0 ; 3655
                                MOVL PKT_POINTER_TAB[R0], PKTPTR
                                TSTL (PKTPTR) ; 3656
                                BEQL 1$
                                0000' CF 9F 0001A PUSHAB KEYWORD_VEC ; 3659
                                50 DD 0001E PUSHL PKTPTR
                                06 DD 00020 PUSHL #6
                                0000V CF 03 FB 00022 CALLS #3, PUT_KEYWORDS
                                0000V CF 01 FB 0002B CALLS #1, STRIP_BLANKS_TRAIL ; 3660
                                09 11 00030 BRB 2$ ; 3656
                                0000' CF 9F 00032 1$: PUSHAB NONE_BUF ; 3663
                                0000V CF 01 FB 00036 CALLS #1, PUT_STRING_ASCII
                                7E D4 0003B 2$: CLRL -(SP) ; 3664
                                0000V CF 01 FB 0003D CALLS #1, PUT_NEWLINE
                                04 00042 RET ; 3665

```

; Routine Size: 67 bytes, Routine Base: \$CODE\$ + 0AB9

```
2392 3666 1 ROUTINE PUT_PKT_SYSUAFF(RECORD_PTR) : NOVALUE =
2393 3667 1
2394 3668 1 ++
2395 3669 1 FUNCTIONAL DESCRIPTION:
2396 3670 1
2397 3671 1 This routine formats a security alarm message line from the
2398 3672 1 first SYSUAFF packet in the security auditing record.
2399 3673 1
2400 3674 1 INPUT:
2401 3675 1 RECORD_PTR : The security alarm record address.
2402 3676 1
2403 3677 1 IMPLICIT INPUT:
2404 3678 1 NONE.
2405 3679 1
2406 3680 1 OUTPUT:
2407 3681 1 NONE.
2408 3682 1
2409 3683 1 IMPLICIT OUTPUT:
2410 3684 1 Formatted message line in the alarm buffer.
2411 3685 1
2412 3686 1 SIDE EFFECTS:
2413 3687 1 NONE.
2414 3688 1
2415 3689 1 ROUTINE VALUE:
2416 3690 1 NONE.
2417 3691 1 --
2418 3692 1
2419 3693 2 BEGIN ! Start of PUT_PKT_SYSUAFF
2420 3694 2
2421 3695 2 LITERAL
2422 3696 2 NUM_SYSUAFF_BITS = 48;
2423 3697 2
2424 3698 2 MAP
2425 3699 2 RECORD_PTR : REF $BBLOCK;
2426 3700 2
2427 3701 2 BIND
2428 3702 2 SYSUAFF_BUF = UPLIT(BYTE(%ASCIC ' Fields Mod: '));
2429 3703 2
2430 3704 2 OWN
2431 3705 2 KEYWORD_VEC : VECTOR[NUM_SYSUAFF_BITS]
2432 3706 2 INITIAL(UPLIT(BYTE(%ASCIC 'ACCESS ')),
2433 3707 2 UPLIT(BYTE(%ASCIC 'ACCOUNT ')),
2434 3708 2 UPLIT(BYTE(%ASCIC 'ASTLM ')),
2435 3709 2 UPLIT(BYTE(%ASCIC 'BATCH ')),
2436 3710 2 UPLIT(BYTE(%ASCIC 'BIOLM ')),
2437 3711 2 UPLIT(BYTE(%ASCIC 'BYTLM ')),
2438 3712 2 UPLIT(BYTE(%ASCIC 'CLI ')),
2439 3713 2 UPLIT(BYTE(%ASCIC 'CLITABLES ')),
2440 3714 2 UPLIT(BYTE(%ASCIC 'CPUTIME ')),
2441 3715 2 UPLIT(BYTE(%ASCIC 'DEFPRIVILEGES ')),
2442 3716 2 UPLIT(BYTE(%ASCIC 'DEVICE ')),
2443 3717 2 UPLIT(BYTE(%ASCIC 'DIALUP ')),
2444 3718 2 UPLIT(BYTE(%ASCIC 'DIOLM ')),
2445 3719 2 UPLIT(BYTE(%ASCIC 'DIRECTORY ')),
2446 3720 2 UPLIT(BYTE(%ASCIC 'ENQLM ')),
2447 3721 2 UPLIT(BYTE(%ASCIC 'EXPIRATION ')),
2448 3722 2 UPLIT(BYTE(%ASCIC 'FILLM ')),
```



```

: 2449      3723  2      UPLIT(BYTE(%ASCIC 'FLAGS      ')),
: 2450      3724  2      UPLIT(BYTE(%ASCIC 'INTERACTIVE  ')),
: 2451      3725  2      UPLIT(BYTE(%ASCIC 'JTQUOTA      ')),
: 2452      3726  2      UPLIT(BYTE(%ASCIC 'LGICMD       ')),
: 2453      3727  2      UPLIT(BYTE(%ASCIC 'LOCAL        ')),
: 2454      3728  2      UPLIT(BYTE(%ASCIC 'MAXDETACH    ')),
: 2455      3729  2      UPLIT(BYTE(%ASCIC 'MAXJOBS      ')),
: 2456      3730  2      UPLIT(BYTE(%ASCIC 'MAXACCTJOBS  ')),
: 2457      3731  2      UPLIT(BYTE(%ASCIC 'NETWORK      ')),
: 2458      3732  2      UPLIT(BYTE(%ASCIC 'OWNER        ')),
: 2459      3733  2      UPLIT(BYTE(%ASCIC 'PASSWORD     ')),
: 2460      3734  2      UPLIT(BYTE(%ASCIC 'PBYTLM       ')),
: 2461      3735  2      UPLIT(BYTE(%ASCIC 'PFLAGS       ')),
: 2462      3736  2      UPLIT(BYTE(%ASCIC 'P_RESTRICT   ')),
: 2463      3737  2      UPLIT(BYTE(%ASCIC 'PGFLQUOTA    ')),
: 2464      3738  2      UPLIT(BYTE(%ASCIC 'PRCLM        ')),
: 2465      3739  2      UPLIT(BYTE(%ASCIC 'PRIMEDAYS    ')),
: 2466      3740  2      UPLIT(BYTE(%ASCIC 'PRIORITY     ')),
: 2467      3741  2      UPLIT(BYTE(%ASCIC 'PRIVILEGES   ')),
: 2468      3742  2      UPLIT(BYTE(%ASCIC 'PWDLIFETIME  ')),
: 2469      3743  2      UPLIT(BYTE(%ASCIC 'PDMINIMUM    ')),
: 2470      3744  2      UPLIT(BYTE(%ASCIC 'QUEPRIORITY  ')),
: 2471      3745  2      UPLIT(BYTE(%ASCIC 'REMOTE       ')),
: 2472      3746  2      UPLIT(BYTE(%ASCIC 'SFLAGS       ')),
: 2473      3747  2      UPLIT(BYTE(%ASCIC 'S_RESTRICT   ')),
: 2474      3748  2      UPLIT(BYTE(%ASCIC 'SHRFILLM     ')),
: 2475      3749  2      UPLIT(BYTE(%ASCIC 'TQELM        ')),
: 2476      3750  2      UPLIT(BYTE(%ASCIC 'UIC          ')),
: 2477      3751  2      UPLIT(BYTE(%ASCIC 'WSDEFAULT    ')),
: 2478      3752  2      UPLIT(BYTE(%ASCIC 'WSEXTENT     ')),
: 2479      3753  2      UPLIT(BYTE(%ASCIC 'WSQUOTA      '))),
: 2480      3754  2
: 2481      3755  2      LOCAL
: 2482      3756  2          PKTPTR      : REF $BBLOCK;
: 2483      3757  2
: 2484      3758  2      PUT STRING ASCIC(SYSUAFF BUF);
: 2485      3759  2      PKTPTR = .PKT POINTER TAB[.PKT_INDEX TAB[NSASK PKTTYP SYSUAFF-1]];
: 2486      3760  2      PUT KEYWORDS(NUM SYSUAFF BITS, PKTPTR[NSASQ_PKT_SYSUAFF], KEYWORD_VEC);
: 2487      3761  2      STRIP BLANKS TRAIL(ALARM_DESCR);
: 2488      3762  2      PUT_NEWLINE(0);
: 2489      3763  1      END;

```

! End of PUT_PKT_SYSUAFF

														.PSECT	\$PLITS, NOWRT, NOEXE, 2					
09	3A	64	6F	4D	20	73	64	6C	65	69	46	09	0D	00670	P.ADM:	.ASCII	<13><9>\Fields Mod:\<9>	:		
														0067E		.BLKB	2			
20	20	20	20	20	20	20	20	53	53	45	43	43	41	0E	00680	P.ADN:	.ASCII	<14>\ACCESS \	:	
														0068F		.BLKB	1			
20	20	20	20	20	20	20	20	54	4E	55	4F	43	43	41	0E	00690	P.ADO:	.ASCII	<14>\ACCOUNT \	:
														0069F		.BLKB	1			
20	20	20	20	20	20	20	20	20	4D	4C	54	53	41	0E	006A0	P.ADP:	.ASCII	<14>\ASTLM \	:	
														006AF		.BLKB	1			
20	20	20	20	20	20	20	20	20	48	43	54	41	42	0E	006B0	P.ADQ:	.ASCII	<14>\BATCH \	:	
														006BF		.BLKB	1			
20	20	20	20	20	20	20	20	20	4D	4C	4F	49	42	0E	006C0	P.ADR:	.ASCII	<14>\BIOLM \	:	

[illegible]


```
20 20 20 20 20 53 59 41 44 45 4D 49 52 50 0E 00890 P.AEU: .ASCII <14>\PRIMEDAYS \
0089F .BLKB 1 ;
20 20 20 20 20 20 59 54 49 52 4F 49 52 50 0E 008A0 P.AEV: .ASCII <14>\PRIORITY \
008AF .BLKB 1 ;
20 20 20 20 53 45 47 45 4C 49 56 49 52 50 0E 008B0 P.AEW: .ASCII <14>\PRIVILEGES \
008BF .BLKB 1 ;
20 20 20 45 4D 49 54 45 46 49 4C 44 57 50 0E 008C0 P.AEX: .ASCII <14>\PWDLIFETIME \
008CF .BLKB 1 ;
20 20 20 20 4D 55 4D 49 4E 49 4D 44 57 50 0E 008D0 P.AEY: .ASCII <14>\PDMINIMUM \
008DF .BLKB 1 ;
20 20 20 59 54 49 52 4F 49 52 50 45 55 51 0E 008E0 P.AEZ: .ASCII <14>\QUEPRIORITY \
008EF .BLKB 1 ;
20 20 20 20 20 20 20 20 45 54 4F 4D 45 52 0E 008F0 P.AFA: .ASCII <14>\REMOTE \
008FF .BLKB 1 ;
20 20 20 20 20 20 20 20 53 47 41 4C 46 53 0E 00900 P.AFB: .ASCII <14>\SFLAGS \
0090F .BLKB 1 ;
20 20 20 20 54 43 49 52 54 53 45 52 5F 53 0E 00910 P.AFC: .ASCII <14>\S_RESTRICT \
0091F .BLKB 1 ;
20 20 20 20 20 20 4D 4C 4C 49 46 52 48 53 0E 00920 P.AFD: .ASCII <14>\SHRFILLM \
0092F .BLKB 1 ;
20 20 20 20 20 20 20 20 20 4D 4C 45 51 54 0E 00930 P.AFE: .ASCII <14>\TQELM \
0093F .BLKB 1 ;
20 20 20 20 20 20 20 20 20 20 20 43 49 55 0E 00940 P.AFF: .ASCII <14>\UIC \
0094F .BLKB 1 ;
20 20 20 20 20 54 4C 55 41 46 45 44 53 57 0E 00950 P.AFG: .ASCII <14>\WSDEFAULT \
0095F .BLKB 1 ;
20 20 20 20 20 20 54 4E 45 54 58 45 53 57 0E 00960 P.AFH: .ASCII <14>\WSEXTENT \
0096F .BLKB 1 ;
20 20 20 20 20 20 20 41 54 4F 55 51 53 57 0E 00970 P.AFI: .ASCII <14>\WSQUOTA \
0097F .BLKB 1 ;
```

.PSECT \$OWNS\$,NOEXE,2

```
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00BD4 KEYWORD_VEC:
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00BEC .ADDRESS P.ADN, P.ADO, P.ADP, P.ADQ, P.ADR, -
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00C04 P.ADS, P.ADT, P.ADU, P.ADV, P.ADW, P.ADX, -
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00C1C P.ADY, P.ADZ, P.AEA, P.AEB, P.AEC, P.AED, -
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00C34 P.AEE, P.AEF, P.AEG, P.AEH, P.AEI, P.AEJ, -
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00C4C P.AEK, P.AEL, P.AEM, P.AEN, P.AEO, P.AEP, -
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00C64 P.AEQ, P.AER, P.AES, P.AET, P.AEU, P.AEV, -
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00C7C P.AEW, P.AEX, P.AEY, P.AEZ, P.AFA, P.AFB, -
P.AFC, P.AFD, P.AFE, P.AFF, P.AFG, P.AFH, -
P.AFI
```

SYSUAFF_BUF= P.ADM

.PSECT \$CODE\$,NOWRT,2

```
0000 00000 PUT_PKT_SYSUAFF:
0000V CF 0000' CF 9F 00002 .WORD Save nothing : 3666
50 0000' 01 FB 00006 PUSHAB SYSUAFF_BUF : 3758
50 0000' CF 3C 0000B CALLS #1, PUT_STRING ASCII :
0000' CF 40 D0 00010 MOVZWL PKT_INDEX TAB+34, R0 : 3759
0000' CF 9F 00016 MOVL PKT_POINTER_TAB[R0], PKTPTR :
50 DD 0001A PUSHAB KEYWORD_VEC : 3760
30 DD 0001C PUSHL PKTPTR :
PUSHL #48 :
```

OPC\$SECURITY
V04-000

C 4
16-Sep-1984 01:48:50
14-Sep-1984 12:50:55

VAX-11 Bliss-32 V4.0-742
[OPCOM.SRC]SECURITY.B32;1

Page 89
(26)

0000V	CF		03	FB	0001E
		0000'	CF	9F	00023
0000V	CF		01	FB	00027
			7E	D4	0002C
0000V	CF		01	FB	0002E
			04	00033	

CALLS	#3, PUT_KEYWORDS
PUSHAB	ALARM_DESCR
CALLS	#1, STRIP_BLANKS_TRAIL
CLRL	-(SP)
CALLS	#1, PUT_NEWLINE
RET	

:	3761
:	3762
:	3763

; Routine Size: 52 bytes, Routine Base: \$CODE\$ + 0AFC


```

: 2491 3764 1 ROUTINE PUT_PKT_PASSWORD(RECORD_PTR) =
: 2492 3765 1
: 2493 3766 1 ++
: 2494 3767 1 FUNCTIONAL DESCRIPTION:
: 2495 3768 1
: 2496 3769 1 This routine formats a security alarm message line from the
: 2497 3770 1 first PASSWORD packet in the security auditing record.
: 2498 3771 1
: 2499 3772 1 INPUT:
: 2500 3773 1 RECORD_PTR : The security alarm record address.
: 2501 3774 1
: 2502 3775 1 IMPLICIT INPUT:
: 2503 3776 1 NONE.
: 2504 3777 1
: 2505 3778 1 OUTPUT:
: 2506 3779 1 NONE.
: 2507 3780 1
: 2508 3781 1 IMPLICIT OUTPUT:
: 2509 3782 1 Formatted message line in the alarm buffer.
: 2510 3783 1
: 2511 3784 1 SIDE EFFECTS:
: 2512 3785 1 NONE.
: 2513 3786 1
: 2514 3787 1 ROUTINE VALUE:
: 2515 3788 1 SSS_NORMAL or a $FAO system service error.
: 2516 3789 1 --
: 2517 3790 1
: 2518 3791 2 BEGIN ! Start of PUT_PKT_PASSWORD
: 2519 3792 2
: 2520 3793 2 MAP
: 2521 3794 2 RECORD_PTR : REF $BBLOCK;
: 2522 3795 2
: 2523 3796 2 BIND
: 2524 3797 2 PASS_BUF = UPLIT(BYTE(%ASCIC ' Password: ')),
: 2525 3798 2 CONTROL = %ASCID '!AF';
: 2526 3799 2
: 2527 3800 2 LOCAL
: 2528 3801 2 PKTPTR : REF $BBLOCK,
: 2529 3802 2 STATUS : LONG,
: 2530 3803 2 LENGTH : LONG;
: 2531 3804 2
: 2532 3805 2 PUT_STRING_ASCIC(PASS_BUF);
: 2533 3806 2 PKTPTR = .PKT_POINTER_TAB[.PKT_INDEX_TAB[NSA$K_PKT_TYP_PASSWORD-1]];
: 2534 3807 2
: 2535 3808 2 ! Move password to message buffer through the FAO AF filter.
: 2536 3809 2
: 2537 P 3810 2 STATUS = $FAO(CONTROL, LENGTH, ALARM_DESCR,
: 2538 3811 2 .PKTPTR[NSA$W_PKT_SIZE] - NSA$K_PKT_HDR_LENGTH, PKTPTR[NSA$T_PKT_DATA]);
: 2539 3812 2 IF NOT .STATUS THEN RETURN .STATUS;
: 2540 3813 2 ALARM_DESCR[DSC$W_LENGTH] = .ALARM_DESCR[DSC$W_LENGTH] - .LENGTH<0,16,0>;
: 2541 3814 2 ALARM_DESCR[DSC$A_POINTER] = .ALARM_DESCR[DSC$A_POINTER] + .LENGTH<0,16,0>;
: 2542 3815 2 PUT_NEWLINE(0);
: 2543 3816 2 RETURN SSS_NORMAL;
: 2544 3817 1 END; ! End of PUT_PKT_PASSWORD
```

```

09 3A 64 72 6F 77 73 73 61 50 09 0B 0097F
00 46 41 21 00980 P.AFJ: .BLKB 1
010E0003 0098C P.AFL: .ASCII <11><9>\Password:\<9>
00000000 00990 P.AFK: .ASCII \!AF\<0>
00000000 00994 P.AFK: .LONG 17694723
00000000 00994 P.AFK: .ADDRESS P.AFL

```

PASS_BUF= P.AFJ
CONTROL= P.AFK

.PSECT \$CODE\$,NOWRT,2

```

0004 00000 PUT_PKT_PASSWORD:
52 0000' CF 9E 00002 .WORD Save R2 : 3764
5E 04 C2 00007 MOVAB ALARM_DESCR, R2
0000V CF 01 9F 0000A SUBL2 #4, SP : 3805
50 30 A2 3C 0000E PUSHAB PASS_BUF
50 44 A240 D0 00013 CALLS #1, PUT_STRING_ASCII : 3806
7E 50 DD 0001C MOVZWL PKT_INDEX_TAB+20, R0
6E FE A0 3C 0001E MOVL PKT_POINTER_TAB[R0], PKTPTR : 3811
0C 52 DD 00025 PUSHL PKTPTR
0000' 04 C2 00022 MOVZWL -2(PKTPTR), -(SP)
00000000G 00 52 DD 00025 SUBL2 #4, (SP)
14 05 FB 00027 PUSHAB LENGTH
62 50 E9 00035 PUSHAB CONTROL
04 50 6E A2 00038 CALLS #5, SYSSFAO : 3812
0000V CF 01 FB 00044 BLBC STATUS, 1$ : 3813
50 01 D0 00049 MOVZWL LENGTH, ALARM_DESCR : 3814
04 A2 50 C0 0003E MOVZWL LENGTH, R0
0000V CF 01 FB 00044 ADDL2 R0, ALARM_DESCR+4 : 3815
50 01 D0 00049 CLRL -(SP)
04 0004C 1$: CALLS #1, PUT_NEWLINE : 3816
RET MOVL #1, R0 : 3817

```

; Routine Size: 77 bytes, Routine Base: \$CODE\$ + 0B30


```
: 2546 3818 1 ROUTINE PUT_PKT_LOGNAM(RECORD_PTR) : NOVALUE =
: 2547 3819 1
: 2548 3820 1 !++
: 2549 3821 1 FUNCTIONAL DESCRIPTION:
: 2550 3822 1
: 2551 3823 1 This routine formats a security alarm message line from the
: 2552 3824 1 first LOGNAM packet in the security auditing record.
: 2553 3825 1
: 2554 3826 1 INPUT:
: 2555 3827 1 RECORD_PTR : The security alarm record address.
: 2556 3828 1
: 2557 3829 1 IMPLICIT INPUT:
: 2558 3830 1 NONE.
: 2559 3831 1
: 2560 3832 1 OUTPUT:
: 2561 3833 1 NONE.
: 2562 3834 1
: 2563 3835 1 IMPLICIT OUTPUT:
: 2564 3836 1 Formatted message line in the alarm buffer.
: 2565 3837 1
: 2566 3838 1 SIDE EFFECTS:
: 2567 3839 1 NONE.
: 2568 3840 1
: 2569 3841 1 ROUTINE VALUE:
: 2570 3842 1 NONE.
: 2571 3843 1 --
: 2572 3844 1
: 2573 3845 2 BEGIN ! Start of PUT_PKT_LOGNAM
: 2574 3846 2
: 2575 3847 2 MAP
: 2576 3848 2 RECORD_PTR : REF $BBLOCK;
: 2577 3849 2
: 2578 3850 2 BIND
: 2579 3851 2 LOGNAM_BUF = UPLIT(BYTE(%ASCIC ' Log Name: '));
: 2580 3852 2
: 2581 3853 2 LOCAL
: 2582 3854 2 PKTPTR : REF $BBLOCK;
: 2583 3855 2
: 2584 3856 2 PUT_STRING_ASCII(LOGNAM_BUF);
: 2585 3857 2 PKTPTR = .PKT_POINTER_TAB[.PKT_INDEX_TAB[NSASK_PKTTP LOGNAM-1]];
: 2586 3858 2 IF (.PKTPTR[NSASK_PKT_SIZE] - NSASK_PKTHDR_LENGTH) GTRU 0
: 2587 3859 2 THEN
: 2588 3860 2 PUT_PKT_STRING(.RECORD_PTR, NSASK_PKTTP_LOGNAM, 1)
: 2589 3861 2 ELSE
: 2590 3862 2 PUT_STRING_ASCII(NONE_BUF);
: 2591 3863 2 PUT_NEWLINE(0);
: 2592 3864 1 END; ! End of PUT_PKT_LOGNAM
```

.PSECT \$PLITS,NOWRT,NOEXE,2

09 3A 65 6D 61 4E 20 67 6F 4C 09 0B 00998 P.AFM: .ASCII <11><9>\Log Name:\<9>

LOGNAM_BUF= P.AFM

```

.PSECT $CODE$,NOWRT,2

0000 00000 PUT_PKT_LOGNAM:
0000V CF 0000' CF 9F 00002 .WORD Save nothing ; 3818
50 01 FB 00006 PUSHAB LOGNAM_BUF ; 3856
50 0000' CF 3C 0000B CALLS #1, PUT_STRING_ASCII ; 3857
50 0000' CF 40 D0 00010 MOVZWL PKT_INDEX_TAB+T0, R0 ; 3858
50 FE A0 3C 00016 MOVZWL PKT_POINTER_TAB[R0], PKTPTR ; 3858
04 04 C2 0001A SUBL2 #4, R0 ; 3860
0E 13 0001D BEQL 1$ ; 3862
01 DD 0001F PUSHL #1 ; 3863
06 DD 00021 PUSHL #6 ; 3864
0000V CF 04 AC DD 00023 PUSHL RECORD_PTR ; 3862
03 FB 00026 CALLS #3, PUT_PKT_STRING ; 3863
09 11 0002B BRB 2$ ; 3864
0000V CF 0000' CF 9F 0002D 1$: PUSHAB NONE_BUF ; 3862
01 FB 00031 CALLS #1, PUT_STRING_ASCII ; 3863
7E D4 00036 2$: CLRL -(SP) ; 3864
0000V CF 01 FB 00038 CALLS #1, PUT_NEWLINE ; 3864
04 0003D RET ; 3864

```

; Routine Size: 62 bytes, Routine Base: \$CODE\$ + 0B7D


```
: 2594 3865 1 ROUTINE PUT_PKT_VOLNAM(RECORD_PTR) : NOVALUE =
: 2595 3866 1
: 2596 3867 1 ++
: 2597 3868 1 FUNCTIONAL DESCRIPTION:
: 2598 3869 1
: 2599 3870 1 This routine formats a security alarm message line from the
: 2600 3871 1 first VOLNAM packet in the security auditing record.
: 2601 3872 1
: 2602 3873 1 INPUT:
: 2603 3874 1 RECORD_PTR : The security alarm record address.
: 2604 3875 1
: 2605 3876 1 IMPLICIT INPUT:
: 2606 3877 1 NONE.
: 2607 3878 1
: 2608 3879 1 OUTPUT:
: 2609 3880 1 NONE.
: 2610 3881 1
: 2611 3882 1 IMPLICIT OUTPUT:
: 2612 3883 1 Formatted message line in the alarm buffer.
: 2613 3884 1
: 2614 3885 1 SIDE EFFECTS:
: 2615 3886 1 NONE.
: 2616 3887 1
: 2617 3888 1 ROUTINE VALUE:
: 2618 3889 1 NONE.
: 2619 3890 1 --
: 2620 3891 1
: 2621 3892 2 BEGIN ! Start of PUT_PKT_VOLNAM
: 2622 3893 2
: 2623 3894 2 MAP
: 2624 3895 2 RECORD_PTR : REF $BBLOCK;
: 2625 3896 2
: 2626 3897 2 LOCAL
: 2627 3898 2 PKTPTR : REF $BBLOCK;
: 2628 3899 2
: 2629 3900 2 BIND
: 2630 3901 2 VOLNAM_BUF = UPLIT(BYTE(%ASCIC ' Vol Name: '));
: 2631 3902 2
: 2632 3903 2 PUT STRING ASCIC(VOLNAM_BUF);
: 2633 3904 2 PKTPTR = .PKT POINTER TAB[.PKT INDEX TAB[NSASK PKTTYP VOLNAM-1]];
: 2634 3905 2 IF (.PKTPTR[NSASK PKT_SIZE] - NSASK PKTHDR_LENGTH) GTRU 0
: 2635 3906 2 THEN
: 2636 3907 2 PUT_PKT_STRING(.RECORD_PTR, NSASK PKTTYP_VOLNAM, 1)
: 2637 3908 2 ELSE
: 2638 3909 2 PUT STRING ASCIC(NONE_BUF);
: 2639 3910 2 PUT_NEWLINE(0);
: 2640 3911 1 END; ! End of PUT_PKT_VOLNAM
```

.PSECT \$PLITS,NOWRT,NOEXE,2

09 3A 65 6D 61 4E 20 6C 6F 56 09 0B 009A4 P.AFN: .ASCII <11><9>\Vol Name:\<9>

VOLNAM_BUF= P.AFN

.PSECT \$CODE\$,NOWRT,2

			0000	00000	PUT_PKT_VOLNAM:			
		0000'	CF	9F	00002	.WORD	Save nothing	: 3865
0000V	CF		01	FB	00006	PUSHAB	VOLNAM_BUF	: 3903
	50	0000'	CF	3C	0000B	CALLS	#1, PUT_STRING_ASCII	
	50	0000'	CF	40	D0	MOVZWL	PKT_INDEX_TAB+T2, R0	: 3904
	50		FE	A0	3C	MOVL	PKT_POINTER_TAB[R0], PKTPTR	
	50			04	C2	MOVZWL	-2(PKTPTR), R0	: 3905
				0E	13	SUBL2	#4, R0	
				01	DD	BEQL	1\$	
				07	DD	PUSHL	#1	: 3907
		04		AC	DD	PUSHL	#7	
0000V	CF			03	FB	PUSHL	RECORD_PTR	
				09	11	CALLS	#3, PUT_PKT_STRING	
		0000'	CF	9F	0002D	BRB	2\$	
0000V	CF		01	FB	00031	PUSHAB	NONE_BUF	: 3909
			7E	D4	00036	CALLS	#1, PUT_STRING_ASCII	
0000V	CF		01	FB	00038	CLRL	-(SP)	: 3910
				04	0003D	CALLS	#1, PUT_NEWLINE	
						RET		: 3911

; Routine Size: 62 bytes, Routine Base: \$CODE\$ + 0BBB


```

: 2642 3912 1 ROUTINE PUT_PKT_VOLSNAM(RECORD_PTR) : NOVALUE =
: 2643 3913 1
: 2644 3914 1 **
: 2645 3915 1 FUNCTIONAL DESCRIPTION:
: 2646 3916 1
: 2647 3917 1 This routine formats a security alarm message line from the
: 2648 3918 1 first VOLSNAM packet in the security auditing record.
: 2649 3919 1
: 2650 3920 1 INPUT:
: 2651 3921 1 RECORD_PTR : The security alarm record address.
: 2652 3922 1
: 2653 3923 1 IMPLICIT INPUT:
: 2654 3924 1 NONE.
: 2655 3925 1
: 2656 3926 1 OUTPUT:
: 2657 3927 1 NONE.
: 2658 3928 1
: 2659 3929 1 IMPLICIT OUTPUT:
: 2660 3930 1 Formatted message line in the alarm buffer.
: 2661 3931 1
: 2662 3932 1 SIDE EFFECTS:
: 2663 3933 1 NONE.
: 2664 3934 1
: 2665 3935 1 ROUTINE VALUE:
: 2666 3936 1 NONE.
: 2667 3937 1 --
: 2668 3938 1
: 2669 3939 2 BEGIN ! Start of PUT_PKT_VOLSNAM
: 2670 3940 2
: 2671 3941 2 MAP
: 2672 3942 2 RECORD_PTR : REF $BBLOCK;
: 2673 3943 2
: 2674 3944 2 BIND
: 2675 3945 2 VOLSNAM_BUF = UPLIT(BYTE(%ASCIC ' Vol Set Name: '));
: 2676 3946 2
: 2677 3947 2 PUT_STRING ASCIC(VOLSNAM_BUF);
: 2678 3948 2 PUT_PKT_STRING(.RECORD_PTR, NSASK_PKTTYP_VOLSNAM, 1);
: 2679 3949 2 PUT_NEWLINE(0);
: 2680 3950 1 END; ! End of PUT_PKT_VOLSNAM
```

```

: 3A 65 6D 61 4E 20 74 65 53 20 6C 6F 56 09 0F 009B0 P.AFO: .ASCII <15><9>\Vol Set Name:\<9>
: 09 009BF
```

VOLSNAM_BUF= P.AFO

.PSECT \$CODE\$,NOWRT,2

```

0000 00000 PUT_PKT_VOLSNAM:
0000V CF 0000' CF 9F 00002 .WORD Save nothing
01 FB 00006 .PUSHAB VOLSNAM_BUF
01 DD 0000B .CALLS #1, PUT_STRING_ASCIC
. PUSHL #1
```

```

: 3912
: 3947
: 3948
```

K 4
16-Sep-1984 01:48:50
14-Sep-1984 12:50:55

Page 97
(30)

		08	DD	0000D	PUSHL	#8
		AC	DD	0000F	PUSHL	RECORD_PTR
0000V	CF	03	FB	00012	CALLS	#3, PUT_PKT_STRING
		7E	D4	00017	CLRL	-(SP)
0000V	CF	01	FB	00019	CALLS	#1, PUT_NEWLINE
			04	0001E	RET	

; Routine Size: 31 bytes, Routine Base: \$CODE\$ + 0BF9


```

: 2682 3951 1 ROUTINE PUT_PKT_VOLPRO(RECORD_PTR) : NOVALUE =
: 2683 3952 1
: 2684 3953 1 ++
: 2685 3954 1 FUNCTIONAL DESCRIPTION:
: 2686 3955 1
: 2687 3956 1 This routine formats a security alarm message line from the
: 2688 3957 1 first VOLPRO packet in the security auditing record.
: 2689 3958 1
: 2690 3959 1 INPUT:
: 2691 3960 1 RECORD_PTR : The security alarm record address.
: 2692 3961 1
: 2693 3962 1 IMPLICIT INPUT:
: 2694 3963 1 NONE.
: 2695 3964 1
: 2696 3965 1 OUTPUT:
: 2697 3966 1 NONE.
: 2698 3967 1
: 2699 3968 1 IMPLICIT OUTPUT:
: 2700 3969 1 Formatted message line in the alarm buffer.
: 2701 3970 1
: 2702 3971 1 SIDE EFFECTS:
: 2703 3972 1 NONE.
: 2704 3973 1
: 2705 3974 1 ROUTINE VALUE:
: 2706 3975 1 NONE.
: 2707 3976 1 --
: 2708 3977 1
: 2709 3978 2 BEGIN ! Start of PUT_PKT_VOLPRO
: 2710 3979 2
: 2711 3980 2 MAP
: 2712 3981 2 RECORD_PTR : REF $BBLOCK;
: 2713 3982 2
: 2714 3983 2 LITERAL
: 2715 3984 2 VOLPRO_BUF_SIZ = 27;
: 2716 3985 2
: 2717 3986 2 BIND
: 2718 3987 2 VOLPRO_BUF = UPLIT(BYTE(%ASCIC ' Vol Pro: '));
: 2719 3988 2
: 2720 3989 2 LOCAL
: 2721 3990 2 PROT_BUF : VECTOR[VOLPRO_BUF_SIZ,BYTE]
: 2722 3991 2 INITIAL(BYTE(%ASCII 'S:RWED,O:RWED,G:RWED,W:RWED')),
: 2723 3992 2
: 2724 3993 2 PROT_DESCR : $BBLOCK[DSC$C_S_BLN]
: 2725 3994 2 PRESET([DSC$W_LENGTH] = VOLPRO_BUF_SIZ,
: 2726 3995 2 [DSC$B_DTYPE] = DSC$K_DTYPE_T,
: 2727 3996 2 [DSC$B_CLASS] = DSC$K_CLASS_D,
: 2728 3997 2 [DSC$A_POINTER] = PROT_BUF),
: 2729 3998 2
: 2730 3999 2 STRING : REF VECTOR[BYTE] INITIAL(PROT_BUF),
: 2731 4000 2 BITS : REF BITVECTOR,
: 2732 4001 2 INDEX1 : LONG,
: 2733 4002 2 INDEX2 : LONG,
: 2734 4003 2 FLAG : LONG;
: 2735 4004 2
: 2736 4005 2 Move the label field into the message buffer and form a pointer to the
: 2737 4006 2 volume protection bits.
: 2738 4007 2

```

```
: 2739      4008 2 PUT_STRING_ASCII(VOLPRO_BUF);
: 2740      4009 2 BITS = .PKT_POINTER_TAB[.PKT_INDEX_TAB[NSASK_PKTTYP_VOLPRO-1]];
: 2741      4010 2
: 2742      4011 2 A bit set indicates no access so for each bit set convert the access
: 2743      4012 2 letter to a blank.
: 2744      4013 2
: 2745      4014 2 INCR INDEX1 FROM 0 TO 3 DO
: 2746      4015 3 BEGIN
: 2747      4016 3 FLAG = 0;
: 2748      4017 3 INCR INDEX2 FROM 0 TO 3 DO
: 2749      4018 4 BEGIN
: 2750      4019 4 IF .BITS[.INDEX1*4+.INDEX2]
: 2751      4020 4 THEN
: 2752      4021 4 PROT_BUF[.INDEX1*7+2+.INDEX2] = ' '
: 2753      4022 4 ELSE
: 2754      4023 4 FLAG = 1;
: 2755      4024 3 END;
: 2756      4025 3 IF NOT .FLAG THEN PROT_BUF[.INDEX1*7+1] = ' ';
: 2757      4026 2 END;
: 2758      4027 2
: 2759      4028 2 Compress the field by removing blanks and move it into the alarm buffer.
: 2760      4029 2
: 2761      4030 2 STRIP_BLANKS_STRING(PROT_DESCR);
: 2762      4031 2 PUT_STRING_ASCII(PROT_DESCR);
: 2763      4032 2 PUT_NEWLINE(0);
: 2764      4033 1 END;

! End of PUT_PKT_VOLPRO
```

```
                                .PSECT $PLITS$,NOWRT,NOEXE,2
                                09 3A 6F 72 50 20 6C 6F 56 09 0A 009C0 P.AFP: .ASCII <10><9>\Vol Pro:\<9>
                                009CB P.AFQ: .BLKB 1
47 2C 44 45 57 52 3A 4F 2C 44 45 57 52 3A 53 009CC P.AFQ: .ASCII \S:RWED,O:RWED,G:RWED,W:RWED\
44 45 57 52 3A 57 2C 44 45 57 52 3A 009DB
```

VOLPRO_BUF= P.AFP

```
                                .PSECT $CODE$,NOWRT,2
                                003C 00000 PUT_PKT_VOLPRO:
                                .WORD Save R2,R3,R4,R5
                                08 AE 0000' 5E 24 C2 00002 .SUBL2 #36, SP
                                CF 1B 28 00005 .MOVC3 #27, P.AFQ, PROT_BUF
                                6E 020E001B 8F D0 0000C .MOVL #34,71963, PROT_DESCR
                                04 AE 08 AE 9E 00013 .MOVAB PROT_BUF, PROT_DESCR+4
                                50 08 AE 9E 00018 .MOVAB PROT_BUF, STRING
                                0000' CF 9F 0001C .PUSHAB VOLPRO_BUF
                                0000V CF 01 FB 00020 .CALLS #1, PUT_STRING_ASCII
                                50 0000' CF 3C 00025 .MOVZWL PKT_INDEX_TAB+24, R0
                                55 0000' CF 40 D0 0002A .MOVL PKT_POINTER_TAB[R0], BITS
                                50 D4 00030 .CLRL INDEX1
                                54 D4 00032 1$: .CLRL FLAG
                                53 50 02 78 00034 .ASHL #2, INDEX1, R3
                                52 D4 00038 .CLRL INDEX2
                                51 53 52 C1 0003A 2$: .ADDL3 INDEX2, R3, R1
```


OE	65	51	E1 0003E	BBC	R1, (BITS), 3\$	:	
51	50	07	C5 00042	MULL3	#7, INDEX1, R1	:	4021
	51	52	C0 00046	ADDL2	INDEX2, R1	:	
	0A AE41	20	90 00049	MOVB	#32, PROT_BUF+2[R1]	:	
		03	11 0004E	BRB	4\$	:	
	54	01	D0 00050 3\$:	MOVL	#1, FLAG	:	4023
E3	52	03	F3 00053 4\$:	AOBLEQ	#3, INDEX2, 2\$	:	4017
	09	54	E8 00057	BLBS	FLAG, 5\$	:	4025
51	50	07	C5 0005A	MULL3	#7, INDEX1, R1	:	
	09 AE41	20	90 0005E	MOVB	#32, PROT_BUF+1[R1]	:	
CB	50	03	F3 00063 5\$:	AOBLEQ	#3, INDEX1, 1\$	:	4014
		5E	DD 00067	PUSHL	SP	:	4030
	0000V CF	01	FB 00069	CALLS	#1, STRIP_BLANKS_STRING	:	
		5E	DD 0006E	PUSHL	SP	:	4031
	0000V CF	01	FB 00070	CALLS	#1, PUT_STRING_ASCID	:	
		7E	D4 00075	CLRL	-(SP)	:	4032
	0000V CF	01	FB 00077	CALLS	#1, PUT_NEWLINE	:	
		04	0007C	RET		:	4033

; Routine Size: 125 bytes, Routine Base: \$CODE\$ + 0C18

```

: 2766      4034 1 ROUTINE PUT_PKT_MOUFLG(RECORD_PTR) : NOVALUE =
: 2767      4035 1
: 2768      4036 1
: 2769      4037 1 ++
: 2770      4038 1 FUNCTIONAL DESCRIPTION:
: 2771      4039 1 This routine formats a security alarm message line from the
: 2772      4040 1 first MOUFLG packet in the security auditing record.
: 2773      4041 1
: 2774      4042 1 INPUT:
: 2775      4043 1 RECORD_PTR : The security alarm record address.
: 2776      4044 1
: 2777      4045 1 IMPLICIT INPUT:
: 2778      4046 1 NONE.
: 2779      4047 1
: 2780      4048 1 OUTPUT:
: 2781      4049 1 NONE.
: 2782      4050 1
: 2783      4051 1 IMPLICIT OUTPUT:
: 2784      4052 1 Formatted message line in the alarm buffer.
: 2785      4053 1
: 2786      4054 1 SIDE EFFECTS:
: 2787      4055 1 NONE.
: 2788      4056 1
: 2789      4057 1 ROUTINE VALUE:
: 2790      4058 1 NONE.
: 2791      4059 1 --
: 2792      4060 1
: 2793      4061 2 BEGIN ! Start of PUT_PKT_MOUFLG
: 2794      4062 2
: 2795      4063 2 LITERAL
: 2796      4064 2 NUM_MOUFLG_BITS = 29;
: 2797      4065 2
: 2798      4066 2 MAP
: 2799      4067 2 RECORD_PTR : REF $BBLOCK;
: 2800      4068 2
: 2801      4069 2 BIND
: 2802      4070 2 MOUFLG_BUF = UPLIT(BYTE(%ASCIC ' Mount Flgs: '));
: 2803      4071 2
: 2804      4072 2 OWN
: 2805      4073 2 KEYWORD_VEC : VECTOR[ NUM MOUFLG BITS]
: 2806      4074 2 INITIAL( UPLIT(BYTE(%ASCIC 'FOREIGN ')),
: 2807      4075 2 UPLIT(BYTE(%ASCIC 'GROUP ')),
: 2808      4076 2 UPLIT(BYTE(%ASCIC 'NOASSIST ')),
: 2809      4077 2 UPLIT(BYTE(%ASCIC 'NODISKQ ')),
: 2810      4078 2 UPLIT(BYTE(%ASCIC 'NOHDR3 ')),
: 2811      4079 2 UPLIT(BYTE(%ASCIC 'NOLABEL ')),
: 2812      4080 2 UPLIT(BYTE(%ASCIC 'NOWRITE ')),
: 2813      4081 2 UPLIT(BYTE(%ASCIC 'OVR_ACCESS ')),
: 2814      4082 2 UPLIT(BYTE(%ASCIC 'OVR_EXP ')),
: 2815      4083 2 UPLIT(BYTE(%ASCIC 'OVR_IDENT ')),
: 2816      4084 2 UPLIT(BYTE(%ASCIC 'OVR_SETID ')),
: 2817      4085 2 UPLIT(BYTE(%ASCIC 'READCHECK ')),
: 2818      4086 2 UPLIT(BYTE(%ASCIC 'SHARE ')),
: 2819      4087 2 UPLIT(BYTE(%ASCIC 'MESSAGE ')),
: 2820      4088 2 UPLIT(BYTE(%ASCIC 'SYSTEM ')),
: 2821      4089 2 UPLIT(BYTE(%ASCIC 'WRITECHECK ')),
: 2822      4090 2 UPLIT(BYTE(%ASCIC 'WRITETHRU ')),

```



```

: 2823      4091  2      UPLIT(BYTE(%ASCIC 'NOCACHE ')),
: 2824      4092  2      UPLIT(BYTE(%ASCIC 'OVR_LOCK ')),
: 2825      4093  2      UPLIT(BYTE(%ASCIC 'NOMNTVER ')),
: 2826      4094  2      UPLIT(BYTE(%ASCIC 'NOUNLOAD ')),
: 2827      4095  2      UPLIT(BYTE(%ASCIC 'NOJRNL ')),
: 2828      4096  2      UPLIT(BYTE(%ASCIC 'NEWJRNL ')),
: 2829      4097  2      UPLIT(BYTE(%ASCIC 'NOAUTO ')),
: 2830      4098  2      UPLIT(BYTE(%ASCIC 'INIT_ALL ')),
: 2831      4099  2      UPLIT(BYTE(%ASCIC 'INIT-CONT ')),
: 2832      4100  2      UPLIT(BYTE(%ASCIC 'OVR_VOLO ')),
: 2833      4101  2      UPLIT(BYTE(%ASCIC 'INTERCHG ')),
: 2834      4102  2      UPLIT(BYTE(%ASCIC 'CLUSTER '));
: 2835      4103  2
: 2836      4104  2      LOCAL
: 2837      4105  2          PKTPTR      : REF $BBLOCK;
: 2838      4106  2
: 2839      4107  2      PUT STRING ASCIC(MOUFLG BUF);
: 2840      4108  2      PKTPTR = .PKT POINTER TAB[.PKT INDEX TAB[NSASK PKTTYP MOUFLG-1]];
: 2841      4109  2      PUT KEYWORDS(NUM MOUFLG BITS, PKTPTR[NSASK_PKT_MOUFLG], KEYWORD_VEC);
: 2842      4110  2      STRIP BLANKS TRAIL(ALARM_DESCR);
: 2843      4111  2      PUT_NEWLINE(0);
: 2844      4112  1      END;

```

! End of PUT_PKT_MOUFLG

.PSECT \$SPLITS,NOWRT,NOEXE,2

09	3A	73	67	6C	46	20	74	6E	75	6F	4D	09	0D	009E7			
														009E7	.BLKB	1	
														009E8	P.AFR:	.ASCII	<13><9>\Mount Flgs:\<9>
														009F6	.BLKB	2	
		20	20	20	20	4E	47	49	45	52	4F	46	0B	009F8	P.AFS:	.ASCII	<11>\FOREIGN \
		20	20	20	20	20	20	50	55	4F	52	47	0B	00A04	P.AFT:	.ASCII	<11>\GROUP \
		20	20	20	54	53	49	53	53	41	4F	4E	0B	00A10	P.AFU:	.ASCII	<11>\NOASSIST \
		20	20	20	20	51	4B	53	49	44	4F	4E	0B	00A1C	P.AFV:	.ASCII	<11>\NODISKQ \
		20	20	20	20	20	33	52	44	48	4F	4E	0B	00A28	P.AFW:	.ASCII	<11>\NOHDR3 \
		20	20	20	20	4C	45	42	41	4C	4F	4E	0B	00A34	P.AFX:	.ASCII	<11>\NOLABEL \
		20	20	20	20	45	54	49	52	57	4F	4E	0B	00A40	P.AFY:	.ASCII	<11>\NOWRITE \
		20	53	53	45	43	43	41	5F	52	56	4F	0B	00A4C	P.AFZ:	.ASCII	<11>\OVR_ACCESS \
		20	20	20	20	50	58	45	5F	52	56	4F	0B	00A58	P.AGA:	.ASCII	<11>\OVR_EXP \
		20	20	54	4E	45	44	49	5F	52	56	4F	0B	00A64	P.AGB:	.ASCII	<11>\OVR_IDENT \
		20	20	44	49	54	45	53	5F	52	56	4F	0B	00A70	P.AGC:	.ASCII	<11>\OVR_SETID \
		20	20	4B	43	45	48	43	44	41	45	52	0B	00A7C	P.AGD:	.ASCII	<11>\READCHECK \
		20	20	20	20	20	20	45	52	41	48	53	0B	00A88	P.AGE:	.ASCII	<11>\SHARE \
		20	20	20	20	45	47	41	53	53	45	4D	0B	00A94	P.AGF:	.ASCII	<11>\MESSAGE \
		20	20	20	20	20	4D	45	54	53	59	53	0B	00AA0	P.AGG:	.ASCII	<11>\SYSTEM \
		20	4B	43	45	48	43	45	54	49	52	57	0B	00AAC	P.AGH:	.ASCII	<11>\WRITECHECK \
		20	20	55	52	48	54	45	54	49	52	57	0B	00AB8	P.AGI:	.ASCII	<11>\WRITETHRU \
		20	20	20	20	45	48	43	41	43	4F	4E	0B	00AC4	P.AGJ:	.ASCII	<11>\NOCACHE \
		20	20	20	4B	43	4F	4C	5F	52	56	4F	0B	00AD0	P.AGK:	.ASCII	<11>\OVR_LOCK \
		20	20	20	52	45	56	54	4E	4D	4F	4E	0B	00ADC	P.AGL:	.ASCII	<11>\NOMNTVER \
		20	20	20	44	41	4F	4C	4E	55	4F	4E	0B	00AE8	P.AGM:	.ASCII	<11>\NOUNLOAD \
		20	20	20	20	20	4C	4E	52	4A	4F	4E	0B	00AF4	P.AGN:	.ASCII	<11>\NOJRNL \
		20	20	20	20	20	4C	4E	52	4A	57	45	0B	00B00	P.AGO:	.ASCII	<11>\NEWJRNL \
		20	20	20	20	20	4F	54	55	41	4F	4E	0B	00B0C	P.AGP:	.ASCII	<11>\NOAUTO \
		20	20	20	4C	4C	41	5F	54	49	4E	49	0B	00B18	P.AGQ:	.ASCII	<11>\INIT_ALL \
		20	20	54	4E	4F	43	5F	54	49	4E	49	0B	00B24	P.AGR:	.ASCII	<11>\INIT-CONT \
		20	20	20	4F	4C	4F	56	5F	52	56	4F	0B	00B30	P.AGS:	.ASCII	<11>\OVR_VOLO \

20 20 20 47 48 43 52 45 54 4E 49 0B 00B3C P.AGT: .ASCII <11>\INTERCHG \
20 20 20 20 52 45 54 53 55 4C 43 0B 00B48 P.AGU: .ASCII <11>\CLUSTER \
.PSECT \$OWNS\$,NOEXE,2

00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00C94 KEYWORD_VEC:
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00CAC .ADDRESS P.AFS, P.AFT, P.AFU, P.AFV, P.AFW, -
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00CC4 P.AFX, P.AFY, P.AFZ, P.AGA, P.AGB, P.AGC, -
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00CDC P.AGD, P.AGE, P.AGF, P.AGG, P.AGH, P.AGI, -
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00CF4 P.AGJ, P.AGK, P.AGL, P.AGM, P.AGN, P.AGO, -
P.AGP, P.AGQ, P.AGR, P.AGS, P.AGT, P.AGU

MOUFLG_BUF= P.AFR

.PSECT \$CODE\$,NOWRT,2

0000 00000 PUT_PKT_MOUFLG:
0000V CF 0000' CF 9F 00002 .WORD Save nothing : 4034
50 01 FB 00006 PUSHAB MOUFLG_BUF : 4107
50 0000' CF 3C 0000B CALLS #1, PUT_STRING_ASCIC :
0000' CF 40 D0 00010 MOVZWL PKT_INDEX_TAB+26, R0 : 4108
0000' CF 9F 00016 MOVL PKT_POINTER_TAB[R0], PKTPTR :
50 DD 0001A PUSHAB KEYWORD_VEC : 4109
1D DD 0001C PUSHL PKTPTR :
0000V CF 03 FB 0001E CALLS #29 :
0000' CF 9F 00023 CALLS #3, PUT_KEYWORDS :
0000V CF 01 FB 00027 PUSHAB ALARM_DESCR : 4110
7E D4 0002C CALLS #1, STRIP_BLANKS_TRAIL :
0000V CF 01 FB 0002E CLRL -(SP) : 4111
04 00033 CALLS #1, PUT_NEWLINE : 4112
RET

; Routine Size: 52 bytes, Routine Base: \$CODE\$ + 0C95


```
: 2846 4113 1 ROUTINE PUT_PKT_DMOUFLG(RECORD_PTR) : NOVALUE =
: 2847 4114 1
: 2848 4115 1 !++
: 2849 4116 1 FUNCTIONAL DESCRIPTION:
: 2850 4117 1
: 2851 4118 1 This routine formats a security alarm message line from the
: 2852 4119 1 first DMOUFLG packet in the security auditing record.
: 2853 4120 1
: 2854 4121 1 INPUT:
: 2855 4122 1 RECORD_PTR : The security alarm record address.
: 2856 4123 1
: 2857 4124 1 IMPLICIT INPUT:
: 2858 4125 1 NONE.
: 2859 4126 1
: 2860 4127 1 OUTPUT:
: 2861 4128 1 NONE.
: 2862 4129 1
: 2863 4130 1 IMPLICIT OUTPUT:
: 2864 4131 1 Formatted message line in the alarm buffer.
: 2865 4132 1
: 2866 4133 1 SIDE EFFECTS:
: 2867 4134 1 NONE.
: 2868 4135 1
: 2869 4136 1 ROUTINE VALUE:
: 2870 4137 1 NONE.
: 2871 4138 1 --
: 2872 4139 1
: 2873 4140 2 BEGIN ! Start of PUT_PKT_DMOUFLG
: 2874 4141 2
: 2875 4142 2 LITERAL
: 2876 4143 2 NUM_DMOUFLG_BITS = 4;
: 2877 4144 2
: 2878 4145 2 MAP
: 2879 4146 2 RECORD_PTR : REF $BBLOCK;
: 2880 4147 2
: 2881 4148 2 BIND
: 2882 4149 2 DMOUFLG_BUF = UPLIT(BYTE(%ASCIC ' Mount Flgs: '));
: 2883 4150 2
: 2884 4151 2 OWN
: 2885 4152 2 KEYWORD_VEC : VECTOR[ NUM_DMOUFLG_BITS ]
: 2886 4153 2 INITIAL( UPLIT( BYTE( %ASCIC 'NOUNLOAD ' ) ),
: 2887 4154 2 UPLIT( BYTE( %ASCIC 'UNIT ' ) ),
: 2888 4155 2 UPLIT( BYTE( %ASCIC 'ABORT ' ) ),
: 2889 4156 2 UPLIT( BYTE( %ASCIC 'CLUSTER' ) ));
: 2890 4157 2
: 2891 4158 2 LOCAL
: 2892 4159 2 PKTPTR : REF $BBLOCK;
: 2893 4160 2
: 2894 4161 2 PUT STRING ASCIC( DMOUFLG_BUF );
: 2895 4162 2 PKTPTR = .PKT POINTER TAB[ .PKT_INDEX TAB[ NSASK PKTTYP DMOUFLG-1 ] ];
: 2896 4163 2 PUT KEYWORDS( NUM_DMOUFLG_BITS, PKTPTR[ NSASK PKT_DMOUFLG ], KEYWORD_VEC );
: 2897 4164 2 STRIP BLANKS TRAIL( ALARM_DESCR );
: 2898 4165 2 PUT_NEWLINE( 0 );
: 2899 4166 1 END; ! End of PUT_PKT_DMOUFLG
```

```

.PSECT $PLITS$,NOWRT,NOEXE,2
09 3A 73 67 6C 46 20 74 6E 75 6F 4D 09 0D 00B54 P.AGV: .ASCII <13><9>\Mount Flgs:\<9> ;
                                00B62 .BLKB 2 ;
                                00B64 P.AGW: .ASCII <9>\NOUNLOAD \ ;
                                00B6E .BLKB 2 ;
                                20 54 49 4E 55 05 00B70 P.AGX: .ASCII <5>\UNIT \ ;
                                00B76 .BLKB 2 ;
                                20 54 52 4F 42 41 06 00B78 P.AGY: .ASCII <6>\ABORT \ ;
                                00B7F .BLKB 1 ;
                                52 45 54 53 55 4C 43 07 00B80 P.AGZ: .ASCII <7>\CLUSTER\ ;

.PSECT $OWNS$,NOEXE,2
00000000' 00000000' 00000000' 00000000' 00D08 KEYWORD_VEC: ;
                                .ADDRESS P.AGW, P.AGX, P.AGY, P.AGZ ;
                                DMOUFLG_BUF= P.AGV ;

.PSECT $CODE$,NOWRT,2
                                0000 00000 PUT_PKT_DMOUFLG:
                                .WORD Save nothing ; 4113
                                PUSHAB DMOUFLG_BUF ; 4161
                                CALLS #1, PUT_STRING_ASCII ;
                                MOVZWL PKT_INDEX_TAB+28, R0 ; 4162
                                MOVL PKT_POINTER_TAB[R0], PKTPTR ;
                                PUSHAB KEYWORD_VEC ; 4163
                                PUSHL PKTPTR ;
                                PUSHL #4 ;
                                CALLS #3, PUT_KEYWORDS ;
                                PUSHAB ALARM_DESCr ; 4164
                                CALLS #1, STRIP_BLANKS_TRAIL ;
                                CLRL -(SP) ; 4165
                                CALLS #1, PUT_NEWLINE ;
                                RET ; 4166

0000V CF 0000' CF 9F 00002
                                01 FB 00006
                                50 0000' CF 3C 0000B
                                50 0000' CF 40 D0 00010
                                0000' CF 9F 00016
                                50 DD 0001A
                                04 DD 0001C
                                0000V CF 03 FB 0001E
                                0000' CF 9F 00023
                                0000V CF 01 FB 00027
                                7E D4 0002C
                                0000V CF 01 FB 0002E
                                04 00033

```

; Routine Size: 52 bytes, Routine Base: \$CODE\$ + 0CC9


```
: 2901      4167 1 ROUTINE PUT_PKT_STRING(RECORD_PTR, PKTTYP, PKTNUM) : NOVALUE =
: 2902      4168 1
: 2903      4169 1 !++
: 2904      4170 1 FUNCTIONAL DESCRIPTION:
: 2905      4171 1
: 2906      4172 1       This routine moves the string data from the Nth packet, (Specified
: 2907      4173 1       by PKTNUM), of type PKTTYP into the alarm buffer.
: 2908      4174 1
: 2909      4175 1 INPUT:
: 2910      4176 1     RECORD_PTR      : The security alarm record address.
: 2911      4177 1     PKTTYP        : The packet type of the data to move.
: 2912      4178 1     PKTNUM        : Specifies which packet of type PKTTYP to use.
: 2913      4179 1
: 2914      4180 1 IMPLICIT INPUT:
: 2915      4181 1     NONE.
: 2916      4182 1
: 2917      4183 1 OUTPUT:
: 2918      4184 1     NONE.
: 2919      4185 1
: 2920      4186 1 IMPLICIT OUTPUT:
: 2921      4187 1     Packet data is moved into the alarm buffer.
: 2922      4188 1
: 2923      4189 1 SIDE EFFECTS:
: 2924      4190 1     NONE.
: 2925      4191 1
: 2926      4192 1 ROUTINE VALUE:
: 2927      4193 1     NONE.
: 2928      4194 1 --
: 2929      4195 1
: 2930      4196 2 BEGIN                                     ! Start of PUT_PKT_STRING
: 2931      4197 2
: 2932      4198 2 MAP
: 2933      4199 2     RECORD_PTR : REF $BBLOCK;
: 2934      4200 2
: 2935      4201 2 LOCAL
: 2936      4202 2     PKTPTR      : REF $BBLOCK,
: 2937      4203 2     ADDRESS     : REF $BBLOCK,
: 2938      4204 2     SIZE        : LONG;
: 2939      4205 2
: 2940      4206 2
: 2941      4207 2 ! Form pointer to packet and get the size and data address.
: 2942      4208 2
: 2943      4209 2 PKTPTR = .PKT_POINTER_TAB[.PKT_INDEX_TAB[PKTTYP-1]+.PKTNUM-1];
: 2944      4210 2 SIZE = .PKTPTR[NSASW_PKT_SIZE] - NSASK_PKTHDR_LENGTH;
: 2945      4211 2 ADDRESS = PKTPTR[NSAST_PRT_DATA];
: 2946      4212 2
: 2947      4213 2 ! The maximum string size that can be placed on one line is defined by
: 2948      4214 2 ! DATA_LINE_SIZE. If the string is longer than DATA_LINE_SIZE it will be
: 2949      4215 2 ! split accross multiple lines.
: 2950      4216 2
: 2951      4217 2 WHILE .SIZE GTRU DATA_LINE_SIZE DO
: 2952      4218 2     BEGIN
: 2953      4219 2
: 2954      4220 2 ! While the size remaining is more than one line of data, move one line
: 2955      4221 2 ! of data followed by a carriage return line feed and three tabs into
: 2956      4222 2 ! the alarm buffer. Adjust the size remaining and data address.
: 2957      4223 2
```

```

: 2958      4224  3      PUT_STRING(DATA_LINE_SIZE, .ADDRESS);
: 2959      4225  3      SIZE = .SIZE - DATA_LINE_SIZE;
: 2960      4226  3      ADDRESS = .ADDRESS + DATA_LINE_SIZE;
: 2961      4227  3      PUT_NEWLINE(1);
: 2962      4228  2      END;
: 2963      4229  2
: 2964      4230  2      ! Move the remainder of the string into the alarm buffer.
: 2965      4231  2
: 2966      4232  2      PUT_STRING(.SIZE, .ADDRESS);
: 2967      4233  1      END;
! End of PUT_PKT_STRING

```

```

                                000C 0000C PUT_PKT_STRING:
                                .WORD      Save R2,R3
50      08 AC D0 00002          MOVL      PKTTP, R0
50      0000'CF40 3C 00006      MOVZWL   PKT_INDEX_TAB-2[R0], R0
50      0C AC C0 0000C          ADDL2    PKTNUM, R0
50      0000'CF40 D0 00010      MOVL      PKT_POINTER_TAB-4[R0], PKTPTR
52      FE A0 3C 00016          MOVZWL   -2(PKTPTR), -SIZE
52      04 C2 0001A          SUBL2     #4, SIZE
53      50 D0 0001D          MOVL      PKTPTR, ADDRESS
37      52 D1 00020 1$:      CML      SIZE, #55
18      1B 00023          BLEQU     2$
53      DD 00025          PUSHL     ADDRESS
37      DD 00027          PUSHL     #55
0000V   CF 02 FB 00029      CALLS     #2, PUT_STRING
52      37 C2 0002E          SUBL2     #55, SIZE
53      37 C0 00031          ADDL2     #55, ADDRESS
0000V   CF 01 DD 00034      PUSHL     #1
01      FB 00036          CALLS     #1, PUT_NEWLINE
E3      11 0003B          BRB       1$
0C      BB 0003D 2$:      PUSHR     #^M<R2,R3>
02      FB 0003F          CALLS     #2, PUT_STRING
04      00044          RET

```

; Routine Size: 69 bytes, Routine Base: \$CODE\$ + 0CFD


```
2969 4234 1 ROUTINE PUT_KEYWORDS(NUM_BITS, BIT_VEC, KEYWORD_VEC) : NOVALUE =
2970 4235 1
2971 4236 1 ++
2972 4237 1 FUNCTIONAL DESCRIPTION:
2973 4238 1
2974 4239 1 This routine translates a bit vector of bits into their keyword
2975 4240 1 equivalents and moves them into the alarm buffer.
2976 4241 1
2977 4242 1 INPUT:
2978 4243 1 NUM_BITS : The number of bits to translate.
2979 4244 1 BIT_VEC : Address of the vector of bits to translate.
2980 4245 1 KEYWORD_VEC : Address of a vector of addresses which point to
2981 4246 1 ASCII keyword strings.
2982 4247 1
2983 4248 1 IMPLICIT INPUT:
2984 4249 1 NONE.
2985 4250 1
2986 4251 1 OUTPUT:
2987 4252 1 NONE.
2988 4253 1
2989 4254 1 IMPLICIT OUTPUT:
2990 4255 1 The keyword equivalents of the bits are moved into the alarm buffer.
2991 4256 1
2992 4257 1 SIDE EFFECTS:
2993 4258 1 NONE.
2994 4259 1
2995 4260 1 ROUTINE VALUE:
2996 4261 1 NONE.
2997 4262 1 --
2998 4263 1
2999 4264 2 BEGIN ! Start of PUT_KEYWORDS
3000 4265 2
3001 4266 2 MAP
3002 4267 2 BIT_VEC : REF BITVECTOR,
3003 4268 2 KEYWORD_VEC : REF VECTOR[,LONG];
3004 4269 2
3005 4270 2 LOCAL
3006 4271 2 INDEX : LONG,
3007 4272 2 COUNT : LONG INITIAL(0),
3008 4273 2 ADDRESS : LONG,
3009 4274 2 SIZE : LONG;
3010 4275 2
3011 4276 2 INCR INDEX FROM 0 TO .NUM_BITS-1 DO
3012 4277 2 BEGIN
3013 4278 2
3014 4279 2 If the bit is set then move the corresponding keyword into
3015 4280 2 the alarm buffer.
3016 4281 2
3017 4282 2 IF .BIT_VEC[.INDEX]
3018 4283 2 THEN
3019 4284 2 BEGIN
3020 4285 2
3021 4286 2 Get the keyword size and address. Update the current line count.
3022 4287 2
3023 4288 2 SIZE = .(.KEYWORD_VEC[.INDEX])<0,8,0>;
3024 4289 2 ADDRESS = .KEYWORD_VEC[.INDEX] + 1;
3025 4290 2 COUNT = .COUNT + .SIZE;
```

```

: 3026      4291  4      |
: 3027      4292  4      |
: 3028      4293  4      | If the addition of this keyword would cause this line to exceed
: 3029      4294  4      | DATA_LINE_SIZE, then move a carriage return line feed and three
: 3030      4295  4      | tabs into the alarm buffer and correct the current line count.
: 3031      4296  4      |
: 3032      4297  4      | IF .COUNT GTRU DATA_LINE_SIZE
: 3033      4298  5      | THEN
: 3034      4299  5      | BEGIN
: 3035      4300  5      | STRIP BLANKS TRAIL(ALARM_DESCR);
: 3036      4301  5      | PUT_NEWLINE(T);
: 3037      4302  4      | COUNT = .SIZE;
: 3038      4303  4      | END;
: 3039      4304  4      |
: 3040      4305  4      | Move the keyword into the alarm buffer.
: 3041      4306  4      |
: 3042      4307  3      | PUT_STRING(.SIZE, .ADDRESS);
: 3043      4308  2      | END;
: 3044      4309  1      | END;

```

! End of PUT_KEYWORDS

003C 00000 PUT_KEYWORDS:

			55	D4	00002	.WORD	Save R2,R3,R4,R5	: 4234
			01	CE	00004	CLRL	COUNT	: 4264
		53	35	11	00007	MNEGL	#1, INDEX	: 4282
			53	E1	00009	BRB	3\$	
30	08	BC	43	D0	0000E	BBC	INDEX, @BIT_VEC, 3\$	
		50	60	9A	00013	MOVL	@KEYWORD_VEC[INDEX], R0	: 4288
		54	A0	9E	00016	MOVZBL	(R0), SIZE	
		52	54	C0	0001A	MOVAB	1(R0), ADDRESS	: 4289
		55	55	D1	0001D	ADDL2	SIZE, COUNT	: 4290
		37	13	1B	00020	CMPL	COUNT, #55	: 4296
			CF	9F	00022	BLEQU	2\$	
			01	FB	00026	PUSHAB	ALARM_DESCR	: 4299
	0000V	CF	01	DD	0002B	CALLS	#1, STRIP_BLANKS_TRAIL	
			01	FB	0002D	PUSHL	#1	: 4300
	0000V	CF	54	D0	00032	CALLS	#1, PUT_NEWLINE	
		55	52	DD	00035	MOVL	SIZE, COUNT	: 4301
			54	DD	00037	PUSHL	ADDRESS	: 4306
			02	FB	00039	PUSHL	SIZE	
C6	0000V	CF	04	AC	F2 0003E	CALLS	#2, PUT_STRING	
		53	04	AC	00043	AOBLSS	NUM_BITS, INDEX, 1\$	: 4276
						RET		: 4309

; Routine Size: 68 bytes, Routine Base: \$CODE\$ + 0D42


```

: 3046 4310 1 ROUTINE PUT_STRING(SIZE, ADDRESS) : NOVALUE =
: 3047 4311 1
: 3048 4312 1 ++
: 3049 4313 1 FUNCTIONAL DESCRIPTION:
: 3050 4314 1
: 3051 4315 1 This routine moves a string specified by SIZE and ADDRESS into the
: 3052 4316 1 alarm buffer.
: 3053 4317 1
: 3054 4318 1 INPUT:
: 3055 4319 1 SIZE : The size of the string to move.
: 3056 4320 1 ADDRESS : The address of the string.
: 3057 4321 1
: 3058 4322 1 IMPLICIT INPUT:
: 3059 4323 1 ALARM_DESCR : The string descriptor of the alarm buffer.
: 3060 4324 1
: 3061 4325 1 OUTPUT:
: 3062 4326 1 NONE.
: 3063 4327 1
: 3064 4328 1 IMPLICIT OUTPUT:
: 3065 4329 1 The string is moved into the alarm buffer and the descriptor is updated.
: 3066 4330 1
: 3067 4331 1 SIDE EFFECTS:
: 3068 4332 1 NONE.
: 3069 4333 1
: 3070 4334 1 ROUTINE VALUE:
: 3071 4335 1 NONE.
: 3072 4336 1 --
: 3073 4337 1
: 3074 4338 2 BEGIN ! Start of PUT_STRING
: 3075 4339 2
: 3076 4340 2 MAP
: 3077 4341 2 SIZE : LONG;
: 3078 4342 2
: 3079 4343 2 CH$MOVE(.SIZE, .ADDRESS, .ALARM_DESCR[DSC$A_POINTER]);
: 3080 4344 2 ALARM_DESCR[DSC$W_LENGTH] = .ALARM_DESCR[DSC$W_LENGTH] - .SIZE;
: 3081 4345 2 ALARM_DESCR[DSC$A_POINTER] = .ALARM_DESCR[DSC$A_POINTER] + .SIZE;
: 3082 4346 1 END; ! End of PUT_STRING

```

003C 00000 PUT_STRING:

0000'	DF	08	BC	04	AC	28	00002	.WORD	Save R2,R3,R4,R5	: 4310
		0000'	CF	04	AC	A2	0000A	MOVC3	SIZE, @ADDRESS, @ALARM_DESCR+4	: 4343
		0000'	CF	04	AC	C0	00010	SUBW2	SIZE, ALARM_DESCR	: 4344
						04	00016	ADDL2	SIZE, ALARM_DESCR+4	: 4345
								RET		: 4346

; Routine Size: 23 bytes, Routine Base: \$CODE\$ + 0D86

```

: 3084 4347 1 ROUTINE PUT_STRING_ASCIC(ASCIC_PTR) : NOVALUE =
: 3085 4348 1
: 3086 4349 1 ++
: 3087 4350 1 FUNCTIONAL DESCRIPTION:
: 3088 4351 1
: 3089 4352 1 This routine moves an ASCIC string specified by ASCIC_PTR into the
: 3090 4353 1 alarm buffer.
: 3091 4354 1
: 3092 4355 1 INPUT:
: 3093 4356 1 ASCIC_PTR : The address of a counted ascii string.
: 3094 4357 1
: 3095 4358 1 IMPLICIT INPUT:
: 3096 4359 1 ALARM_DESCR : The string descriptor of the alarm buffer.
: 3097 4360 1
: 3098 4361 1 OUTPUT:
: 3099 4362 1 NONE.
: 3100 4363 1
: 3101 4364 1 IMPLICIT OUTPUT:
: 3102 4365 1 The string is moved into the alarm buffer and the alarm
: 3103 4366 1 descriptor is updated.
: 3104 4367 1
: 3105 4368 1 SIDE EFFECTS:
: 3106 4369 1 NONE.
: 3107 4370 1
: 3108 4371 1 ROUTINE VALUE:
: 3109 4372 1 NONE.
: 3110 4373 1 --
: 3111 4374 1
: 3112 4375 2 BEGIN ! Start of PUT_STRING_ASCIC
: 3113 4376 2
: 3114 4377 2 MAP
: 3115 4378 2 ASCIC_PTR : REF VECTOR[.BYTE];
: 3116 4379 2
: 3117 4380 2 CH$MOVE(.ASCIC_PTR[0], ASCIC_PTR[1], .ALARM_DESCR[DSC$A_POINTER]);
: 3118 4381 2 ALARM_DESCR[DSC$W_LENGTH] = .ALARM_DESCR[DSC$W_LENGTH] + .ASCIC_PTR[0];
: 3119 4382 2 ALARM_DESCR[DSC$A_POINTER] = .ALARM_DESCR[DSC$A_POINTER] + .ASCIC_PTR[0];
: 3120 4383 1 END; ! End of PUT_STRING_ASCIC

```

007C 0000 PUT_STRING_ASCIC:

		56	04	AC	D0	00002	.WORD	Save R2,R3,R4,R5,R6	: 4347
		50		66	9A	00006	MOVL	ASCIC_PTR, R6	: 4380
0000'	DF	01		50	28	00009	MOVZBL	(R6), R0	
		50		66	9A	00010	MOVZBL	(R6), R0	: 4381
		CF		50	A2	00013	SUBW2	R0, ALARM_DESCR	: 4382
		50		66	9A	00018	MOVZBL	(R6), R0	: 4383
		CF		50	C0	0001B	ADDL2	R0, ALARM_DESCR+4	
				04	00020		RET		

; Routine Size: 33 bytes, Routine Base: \$CODE\$ + 0D9D


```

: 3122 4384 1 ROUTINE PUT_STRING_ASCID(DESCR_PTR) : NOVALUE =
: 3123 4385 1
: 3124 4386 1 ++
: 3125 4387 1 FUNCTIONAL DESCRIPTION:
: 3126 4388 1
: 3127 4389 1 This routine moves a string specified by DESCR_PTR into the
: 3128 4390 1 alarm buffer.
: 3129 4391 1
: 3130 4392 1 INPUT:
: 3131 4393 1 DESCR_PTR : Address of the string descriptor.
: 3132 4394 1
: 3133 4395 1 IMPLICIT INPUT:
: 3134 4396 1 ALARM_DESCR : The string descriptor of the alarm buffer.
: 3135 4397 1
: 3136 4398 1 OUTPUT:
: 3137 4399 1 NONE.
: 3138 4400 1
: 3139 4401 1 IMPLICIT OUTPUT:
: 3140 4402 1 The string is moved into the alarm buffer and the alarm descriptor
: 3141 4403 1 is updated.
: 3142 4404 1
: 3143 4405 1 SIDE EFFECTS:
: 3144 4406 1 NONE.
: 3145 4407 1
: 3146 4408 1 ROUTINE VALUE:
: 3147 4409 1 NONE.
: 3148 4410 1 --
: 3149 4411 1
: 3150 4412 2 BEGIN ! Start of PUT_STRING_ASCID
: 3151 4413 2
: 3152 4414 2 MAP
: 3153 4415 2 DESCR_PTR : REF $BLOCK;
: 3154 4416 2
: 3155 4417 2 CH$MOVE(.DESCR_PTR[DSC$W_LENGTH], .DESCR_PTR[DSC$A_POINTER], .ALARM_DESCR[DSC$A_POINTER]);
: 3156 4418 2 ALARM_DESCR[DSC$W_LENGTH] = .ALARM_DESCR[DSC$W_LENGTH] - .DESCR_PTR[DSC$W_LENGTH];
: 3157 4419 2 ALARM_DESCR[DSC$A_POINTER] = .ALARM_DESCR[DSC$A_POINTER] + .DESCR_PTR[DSC$W_LENGTH];
: 3158 4420 1 END; ! End of PUT_STRING_ASCID

```

007C 00000 PUT_STRING_ASCID:

						.WORD	Save R2,R3,R4,R5,R6	: 4384
						MOVL	DESCR_PTR, R6	: 4417
0000'	DF	04	B6	04	AC D0 00002	MOVCL	(R6), @4(R6), @ALARM_DESCR+4	: 4418
		0000'	CF		66 A2 0000D	SUBW2	(R6), ALARM_DESCR	: 4419
		0000'	50		66 3C 00012	MOVZWL	(R6), R0	: 4420
			CF		50 C0 00015	ADDL2	R0, ALARM_DESCR+4	
					04 0001A	RET		

; Routine Size: 27 bytes, Routine Base: \$CODE\$ + 0DBE

```

: 3160 4421 1 ROUTINE PUT_NEWLINE(FLAGS) : NOVALUE =
: 3161 4422 1
: 3162 4423 1 ++
: 3163 4424 1 FUNCTIONAL DESCRIPTION:
: 3164 4425 1
: 3165 4426 1 This routine moves a carriage return, line feed, and optionally 3 tabs
: 3166 4427 1 into the alarm buffer.
: 3167 4428 1
: 3168 4429 1 INPUT:
: 3169 4430 1     FLAGS           : When 0 only the CRLF is put in the buffer. If 1, a
: 3170 4431 1                  CRLF and three tabs are moved into the alarm buffer.
: 3171 4432 1
: 3172 4433 1 IMPLICIT INPUT:
: 3173 4434 1     ALARM_DESCR      : The string descriptor of the alarm buffer.
: 3174 4435 1
: 3175 4436 1 OUTPUT:
: 3176 4437 1     NONE.
: 3177 4438 1
: 3178 4439 1 IMPLICIT OUTPUT:
: 3179 4440 1     The data is moved into the alarm buffer and the alarm descriptor
: 3180 4441 1 is updated.
: 3181 4442 1
: 3182 4443 1 SIDE EFFECTS:
: 3183 4444 1     NONE.
: 3184 4445 1
: 3185 4446 1 ROUTINE VALUE:
: 3186 4447 1     NONE.
: 3187 4448 1 --
: 3188 4449 1
: 3189 4450 2 BEGIN                               ! Start of PUT_NEWLINE
: 3190 4451 2
: 3191 4452 2 OWN
: 3192 4453 2     CRLFTABS      : VECTOR[5,BYTE] INITIAL(BYTE(13,10,9,9,9));
: 3193 4454 2
: 3194 4455 2 LOCAL
: 3195 4456 2     SIZE          : LONG;
: 3196 4457 2
: 3197 4458 2 IF .FLAGS
: 3198 4459 2 THEN
: 3199 4460 2     SIZE = 5
: 3200 4461 2 ELSE
: 3201 4462 2     SIZE = 2;
: 3202 4463 2 CH$MOVE(.SIZE, CRLFTABS, .ALARM_DESCR[DSC$A_POINTER]);
: 3203 4464 2 ALARM_DESCR[DSC$W_LENGTH] = .ALARM_DESCR[DSC$W_LENGTH] - .SIZE;
: 3204 4465 2 ALARM_DESCR[DSC$A_POINTER] = .ALARM_DESCR[DSC$A_POINTER] + .SIZE;
: 3205 4466 1 END;                               ! End of PUT_NEWLINE

```

.PSECT \$OWNS,NOEXE,2

09 09 09 0A 0D 00D18 CRLFTABS:
.BYTE 13, 10, 9, 9, 9 ;

.PSECT \$CODE\$,NOWRT,2

				007C 00000	PUT_NEWLINE:			
					.WORD	Save R2,R3,R4,R5,R6	:	4421
		05	04	AC	E9 00002	BLBC	FLAGS, 1\$	: 4458
		56		05	D0 00006	MOVL	#5, SIZE	: 4460
				03	11 00009	BRB	2\$	:
		56		02	D0 0000B	1\$: MOVL	#2, SIZE	: 4462
0000'	DF	0000'	CF	56	28 0000E	2\$: MOVC3	SIZE, CRLFTABS, @ALARM_DESCR+4	: 4463
		0000'	CF	56	A2 00016	SUBW2	SIZE, ALARM_DESCR	: 4464
		0000'	CF	56	C0 0001B	ADDL2	SIZE, ALARM_DESCR+4	: 4465
				04	00020	RET		: 4466

```
; Routine Size: 33 bytes,    Routine Base: $CODE$ + 0DD9
```

```

: 3207      4467 1 ROUTINE STRIP_BLANKS_TRAIL(DESCR_PTR) : NOVALUE =
: 3208      4468 1
: 3209      4469 1 ++
: 3210      4470 1 FUNCTIONAL DESCRIPTION:
: 3211      4471 1
: 3212      4472 1     This routine removes trailing blanks from the string described
: 3213      4473 1     by DESCR_PTR.
: 3214      4474 1
: 3215      4475 1 INPUT:
: 3216      4476 1     DESCR_PTR      : The address of a string descriptor.
: 3217      4477 1
: 3218      4478 1 IMPLICIT INPUT:
: 3219      4479 1     NONE.
: 3220      4480 1
: 3221      4481 1 OUTPUT:
: 3222      4482 1     NONE.
: 3223      4483 1
: 3224      4484 1 IMPLICIT OUTPUT:
: 3225      4485 1     Trailing blanks are removed from the string described by DESCR_PTR and
: 3226      4486 1     the descriptor size field is updated.
: 3227      4487 1
: 3228      4488 1 SIDE EFFECTS:
: 3229      4489 1     NONE.
: 3230      4490 1
: 3231      4491 1 ROUTINE VALUE:
: 3232      4492 1     NONE.
: 3233      4493 1 --
: 3234      4494 1
: 3235      4495 2 BEGIN                                ! Start of STRIP_BLANKS_TRAIL
: 3236      4496 2
: 3237      4497 2 MAP
: 3238      4498 2     DESCR_PTR      : REF $BBLOCK;
: 3239      4499 2
: 3240      4500 2 LOCAL
: 3241      4501 2     INDEX          : LONG INITIAL(0);
: 3242      4502 2
: 3243      4503 2 WHILE (.DESCR_PTR[DSC$A_POINTER] - 1 - .INDEX) < 0, 8, 0> EQLU ' ' DO INDEX = .INDEX + 1;
: 3244      4504 2 DESCR_PTR[DSC$W_LENGTH] = .DESCR_PTR[DSC$W_LENGTH] + .INDEX;
: 3245      4505 2 DESCR_PTR[DSC$A_POINTER] = .DESCR_PTR[DSC$A_POINTER] - .INDEX;
: 3246      4506 1 END;                                ! End of STRIP_BLANKS_TRAIL

```

```

                                0004 00000 STRIP_BLANKS_TRAIL:
                                .WORD Save R2
                                CLRL INDEX
                                MOVL DESCR_PTR, R0
51      04      50      04      52      D4 00002      1$:      SUBL3 INDEX, 4(R0), R1
                                52      C3 00008
                                FF      A1 91 0000D      CMPB -1(R1), #32
                                04      12 00011      BNEQ 2$
                                52      D6 00013      INCL INDEX
                                F1      11 00015      BRB 1$
                                04      BC      52      A0 00017 2$:      ADDW2 INDEX, @DESCR_PTR
                                04      A0      52      C2 0001B      SUBL2 INDEX, 4(R0)
                                04      0001F      RET
                                : 4467
                                : 4495
                                : 4503
                                :
                                :
                                :
                                :
                                : 4504
                                : 4505
                                : 4506

```


OPCSSECURITY
V04-000

D 6
16-Sep-1984 01:48:50
14-Sep-1984 12:50:55

VAX-11 Bliss-32 V4.0-742
[OPCOM.SRC]SECURITY.B32;1

Page 116
(40)

; Routine Size: 32 bytes, Routine Base: \$CODES + ODFA

```

: 3248 4507 1 ROUTINE STRIP_BLANKS_STRING(DESCR_PTR) : NOVALUE =
: 3249 4508 1
: 3250 4509 1 !++
: 3251 4510 1 FUNCTIONAL DESCRIPTION:
: 3252 4511 1
: 3253 4512 1 This routine removes blanks and compresses the string described
: 3254 4513 1 by DESCR_PTR.
: 3255 4514 1
: 3256 4515 1 INPUT:
: 3257 4516 1 DESCR_PTR : The address of a string descriptor.
: 3258 4517 1
: 3259 4518 1 IMPLICIT INPUT:
: 3260 4519 1 NONE.
: 3261 4520 1
: 3262 4521 1 OUTPUT:
: 3263 4522 1 NONE.
: 3264 4523 1
: 3265 4524 1 IMPLICIT OUTPUT:
: 3266 4525 1 Blanks are removed and the string is compressed. The descriptor
: 3267 4526 1 size field is updated.
: 3268 4527 1
: 3269 4528 1 SIDE EFFECTS:
: 3270 4529 1 NONE.
: 3271 4530 1
: 3272 4531 1 ROUTINE VALUE:
: 3273 4532 1 NONE.
: 3274 4533 1 --
: 3275 4534 1
: 3276 4535 2 BEGIN ! Start of STRIP_BLANKS_STRING
: 3277 4536 2
: 3278 4537 2 MAP
: 3279 4538 2 DESCR_PTR : REF $BBLOCK;
: 3280 4539 2
: 3281 4540 2 LOCAL
: 3282 4541 2 STRING : REF VECTOR[.BYTE] INITIAL(.DESCR_PTR[DSC$A_POINTER]),
: 3283 4542 2 BLANK_COUNT : INITIAL(0),
: 3284 4543 2 INDEX : LONG;
: 3285 4544 2
: 3286 4545 2 INCR INDEX FROM 0 TO .DESCR_PTR[DSC$W_LENGTH]-1 DO
: 3287 4546 3 BEGIN
: 3288 4547 3 IF .STRING[.INDEX] EQLU ' '
: 3289 4548 3 THEN
: 3290 4549 3 BLANK_COUNT = .BLANK_COUNT + 1
: 3291 4550 3 ELSE
: 3292 4551 3 STRING[.INDEX-.BLANK_COUNT] = .STRING[.INDEX];
: 3293 4552 2 END;
: 3294 4553 2 DESCR_PTR[DSC$W_LENGTH] = .DESCR_PTR[DSC$W_LENGTH] - .BLANK_COUNT;
: 3295 4554 1 END; ! End of STRIP_BLANKS_STRING

```

003C 00000 STRIP_BLANKS_STRING:

52	04	AC	DO	00002	.WORD	Save R2,R3,R4,R5	: 4507
51	04	A2	DO	00006	MOVL	DESCR_PTR, R2	: 4541
					MOVL	4(R2), STRING	:

		55	D4	0000A	CLRL	BLANK_COUNT	
54		62	3C	0000C	MOVZWL	(R2), R4	4545
50		01	CE	0000F	MNEGL	#1, INDEX	4547
		13	11	00012	BRB	3\$	
20		6041	91	00014	1\$: CMPB	(INDEX)[STRING], #32	
		04	12	00018	BNEQ	2\$	
		55	D6	0001A	INCL	BLANK_COUNT	4549
		09	11	0001C	BRB	3\$	
53	50	55	C3	0001E	2\$: SUBL3	BLANK_COUNT, INDEX, R3	4551
	6341	6041	90	00022	MOVB	(INDEX)[STRING], (R3)[STRING]	
E9	50	54	F2	00027	3\$: AOBLS	R4, INDEX, 1\$	4545
	62	55	A2	0002B	SUBW2	BLANK_COUNT, (R2)	4553
		04	0002E	RET			4554

; Routine Size: 47 bytes, Routine Base: \$CODE\$ + 0E1A

; 3296 4555 1
; 3297 4556 1 END
; 3298 4557 0 ELUDOM

! End of SECURITY

PSECT SUMMARY

Name	Bytes	Attributes
\$SPLITS	2952	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$OWNS	3357	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	3657	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	125	0	1000	00:01.8
\$255\$DUA28:[OPCOM.OBJ]OPCOMLIB.L32;1	633	22	3	43	00:00.7

COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD, INITIAL, OPTIMIZE)/LIS=LISS:SECURITY/OBJ=OBJ\$:SECURITY MSRC\$:SECURITY/UPDATE=(ENH\$:SECURITY)

; Size: 3657 code + 6309 data bytes
; Run Time: 01:27.6

OPCSSECURITY
V04-000

G 6
16-Sep-1984 01:48:50

VAX-11 Bliss-32 V4.0-742

Page 119

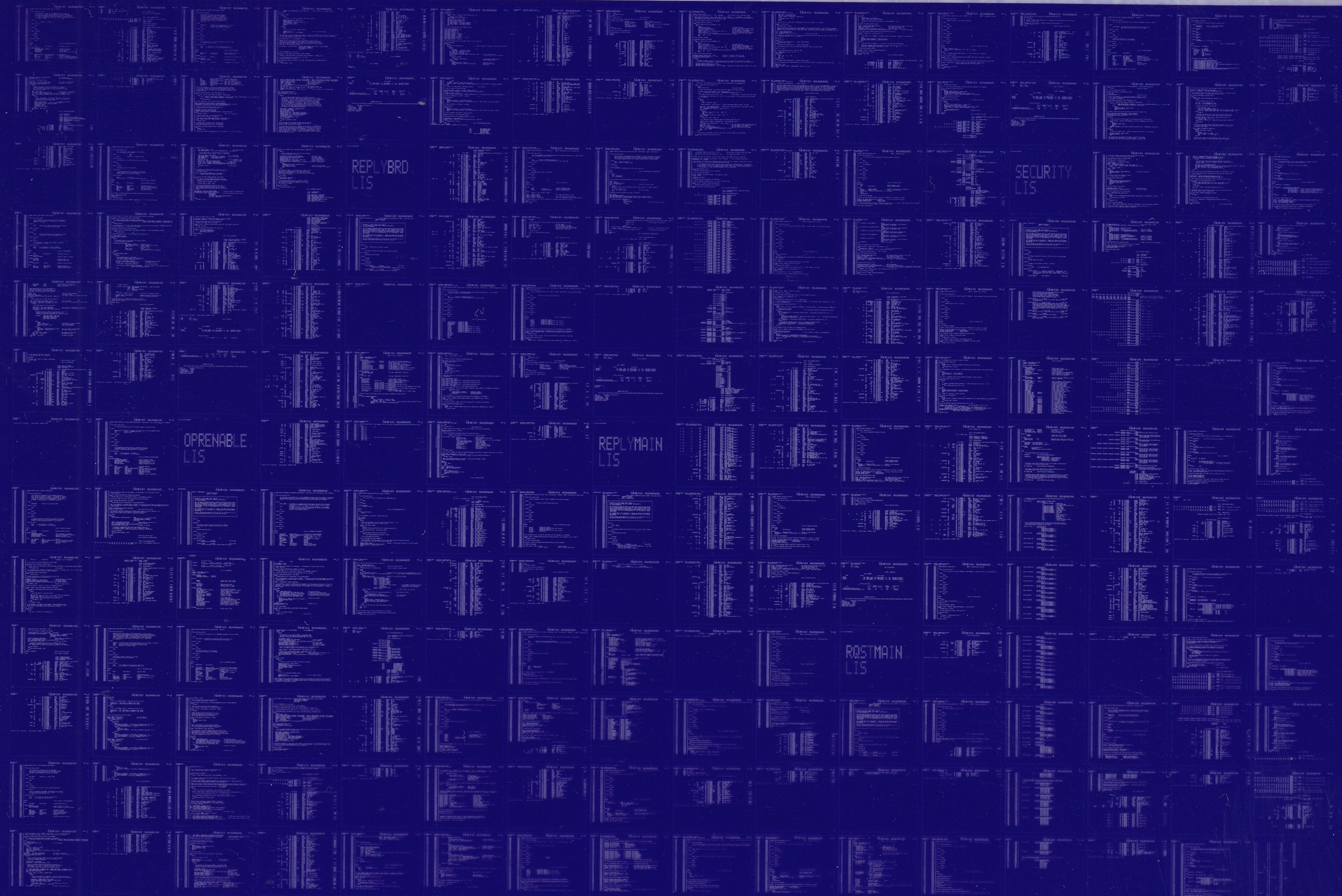
; Elapsed Time: 04:41.4
; Lines/CPU Min: 3122
; Lexemes/CPU-Min: 19733
; Memory Used: 372 pages
; Compilation Complete

OP
VO

00
00
00
00

0291 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0292 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY